

Senior Center Care Database

Bingchun Chang, Rediet Tegegne, Weiwei Liu, Yifei Li

CS5200: Northeastern University, Boston, MA, USA

Abstract

The Senior Center Care Database (SCCD) is a health care tool we developed to facilitate the communication and information exchange of geriatric health care providers and senior living facility staff. The overarching goal of this project and tool is to help our stakeholders - the health care providers and the living facility staff, have a go-to tool, that provides information such as the drugs a patient is prescribed, the frequency of administration, the ailments they may have, potential co-morbidities to watch out for, how often the patient needs to visit certain types of doctors, contact information for the family of the patient and who are due for a routine check based on their last checkup date... Moreover, the database is anonymized via alphanumeric patient IDs.

Introduction

Elderly, post-retirement adults are a vulnerable population, facing health and lifestyle challenges. Many choose to live in an assisted living facility, which is a type of housing that provides health care and in-house nurses, that care for the residents and facilitates their needs such as coordinating doctors' visits, administering medication, and checking in with the resident's family as needed. The SCCD provides a single source for all the above mentioned, with data regarding the residents' current medication needs, the email and phone numbers of their family and/or emergency contact(s), the cadence of their medical appointments, their existing ailments, such that the center could plan preventative screenings for any co-morbidities they may suspect. Additional functionality that could be built in is drug interaction screening, where if a resident is prescribed a new drug, the staff could check it against the main known drug interactions in the database (eg: patient is prescribed statins, the staff member gets a note that says 'avoid grapefruit').

Data Sources

The Senior Center Care Database (SCCD) is a database application developed to streamline communication and information exchange among geriatric healthcare providers and staff at senior living facilities. Due to the sensitive nature of health data, real patient information was not accessible for this project. Instead, we utilized www.mockaroo.com to generate mock data that encompasses all relevant database entities.

Database Design

The SCCD Database is structured to address the primary needs of residents and the daily operational requirements of staff members. It incorporates essential information such as emergency contacts, family details, special care needs, and critical health data including active prescriptions and upcoming appointments. The database schema employs a relational model with numerous one-to-many relationships to ensure comprehensive coverage of resident-related information.

The ER Diagram (*Figure 1*), created using [Draw.io](https://draw.io), illustrates the schema's structure, while *Figure 2* provides a detailed overview of all entities within the database along with concise descriptions and their respective relationships.

Figure 1: SCCD ER Diagram

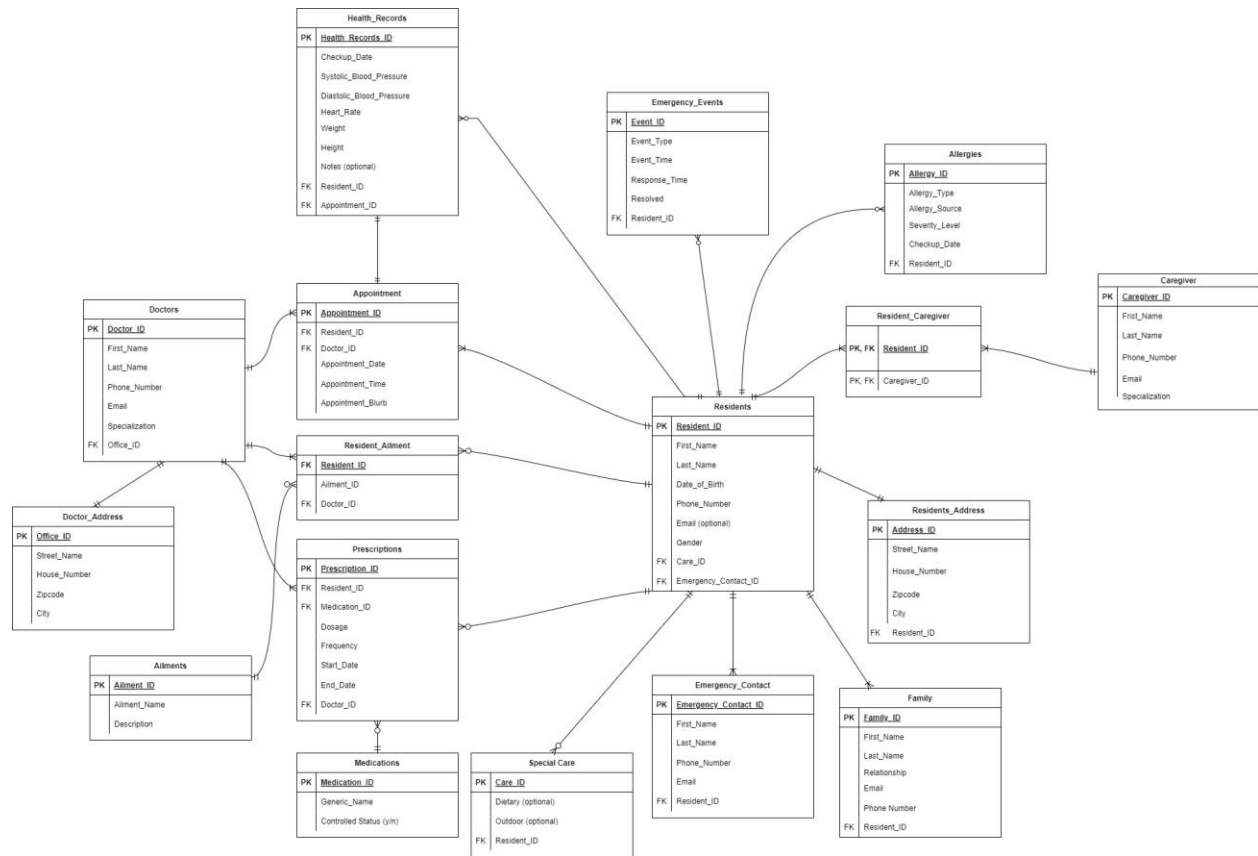


Figure 2: SCCD Detailed Entity Descriptions

Table Name	Brief Description	Summary of Relations
Residents	Stores information about residents (Name, DOB etc...)	Related to Appointments, Emergency_Events, Health_Records, Prescriptions, Allergies, Addresses, Caregivers, Family, and Emergency Contacts tables
Appointments	Tracks residents' appointments	Linked to Residents, Doctors, and Resident_Ailment tables

Health_Records	Contains health records of residents (Blood Pressure, Heart rate etc...)	Linked to Residents and Appointments tables
Prescriptions	Lists prescriptions given to residents	Linked to Residents, Doctors, and Medications tables
Allergies	Allergy information of residents	Linked to Residents table
Doctors	Stores information about doctors	Linked to Appointments and Resident_Ailment tables
Ailments	Contains information about various ailments	Linked to Resident_Ailment table
Resident_Ailment	Tracks the various ailments/diseases/conditions residents may have	Linked to Residents, Doctors, and Ailments tables
Medications	Contains details of medications (Name and [Y/N] on controlled status)	Linked to Prescriptions table
Caregivers	Stores information about the residents' caregivers (Name, contact info, specialization etc...)	Linked to Residents table
Emergency_Events	Records emergency events related to residents (Date, Time, Resolved [Y/N])	Linked to Residents table
Emergency_Contact	Contains emergency contact information for residents	Linked to Residents table
Family	Stores information about family members of residents	Linked to Residents table
Residents_Address	Contains address details of residents (Street, ZipCode etc...)	Linked to Residents table
Doctor_Address	Contains address details of doctor offices (Street, ZipCode etc...)	Linked to Doctors table
Special_Care	Contains information about special care requirements of residents (Daily sunlight, walks etc...)	Linked to Residents table

Data Sources and Methods

The SCCD database leverages mock data generated from Mockaroo.com. All schemas and relationships are replicated on Mockaroo, followed by the generation of SQL insert statements and CSV data files for database integration.

To simulate a senior care center, the database includes 35 residents along with their families and emergency contacts. Additionally, it features 12 doctors and 15 caregivers, with each doctor and caregiver managing multiple residents. Appointments are conducted by a doctor and

attended by a resident. Upon completion of an appointment, the procedure ***InsertHealthRecordForAppointment*** generates a health record for that appointment.

For specific categories such as ailments, allergies, medications, and special care needs, a limited number of generic types are used for simplicity.

Data entries are tailored to reflect real-world scenarios. For instance, the residents' dates of birth are constrained to a realistic human lifespan.

However, Mockaroo has some limitations. It does not support pattern matching or referencing data objects across tables, leading to occasional discrepancies. For example, an allergy type of "peanut" may erroneously have an allergy source of "pet dander," and a resident's height or weight may vary significantly across different health records due to the lack of data referencing between rows.

These mismatches are mainly found in tables requiring resident-specific customizations and frequent updates, such as resident-specific allergy records and post-appointment health records. Despite these oddities, the overall database structure and data integrity remain sound, effectively demonstrating the SCCD application's full functionalities.

User Cases (Application Prototype)

The following examples demonstrate various use-cases for the database:

1. The Senior Center is hiring a new Van/Transport driver to shuttle residents to and from appointments, use the SCCD to generate a list of appointments that occur on a specific date (Snippet for 2021-12-12), and list the Appointment time, the zip code of the doctor's office, and the street address of the doctor's office. Order by Appointment time and then zip code to facilitate the driver's route planning

Query

```
SELECT
    a.Appointment_Time AS AppointmentTime,
    da.Zipcode,
    CONCAT(da.Street_Name, ' ', da.House_Number) AS StreetAddress,
    CONCAT(r.First_Name, ' ', r.Last_Name) AS ResidentName
FROM
    Appointment a
JOIN
    Residents r ON a.Resident_ID = r.Resident_ID
JOIN
    Doctors d ON a.Doctor_ID = d.Doctor_ID
JOIN
    Doctor_Address da ON d.Office_ID = da.Office_ID
WHERE
    a.Appointment_Date = '2021-12-12' -- testing with a specific date
ORDER BY
    a.Appointment_Time, da.Zipcode;
```

Tabular Output

	AppointmentTime	Zipcode	StreetAddress	ResidentName
▶	11:46:00	1415	64 Old Gate Parkway PO Box 92355	Murdoch Dobie

2. Generate a list of residents who need special dietary meals delivered for a given day, sorted by dietary requirement and resident's address.

Query

```

SELECT
    r.First_Name AS ResidentFirstName,
    r.Last_Name AS ResidentLastName,
    sc.Dietary AS DietaryRequirement,
    CONCAT(ra.Street_Name, ' ', ra.House_Number) AS StreetAddress,
    ra.Zipcode,
    ra.City
FROM
    Residents r
JOIN
    Special_Care sc ON r.Care_ID = sc.Care_ID
JOIN
    Residents_Address ra ON r.Resident_ID = ra.Resident_ID
WHERE
    sc.Dietary IS NOT NULL
ORDER BY
    sc.Dietary, ra.Zipcode, ra.Street_Name, ra.House_Number;

```


Tabular Output:

	ResidentFirstName	ResidentLastName	DietaryRequirement	StreetAddress	Zipcode	City
►	Uri	Van der Brugge	dairy-free	21961 Lakeland Center Suite 40	1415	Jebba
	Lewiss	Morehall	dairy-free	61 Ronald Regan Park Room 1643	421 10	Västra Frölunda
	Elbert	Twinterman	dairy-free	841 Ridge Oak Court Suite 14	46202	Indianapolis
	Murdoch	Dobie	dairy-free	58 Commercial Road Room 1153	9101	Vecumnieki
	Arlene	Youster	diabetic	5 Reinke Terrace Suite 93	1234	Kalembutillu
	Teresina	Eilhertsen	diabetic	50 Alpine Drive PO Box 61066	2021	Changleng
	Michell	Jean	diabetic	6925 Division Alley 8th Floor	2223	Sepulu
	Gibb	Chappell	diabetic	4 Morningstar Place 17th Floor	25610	Kuantan
	Donnie	Kezourec	diabetic	58 Clarendon Circle Room 292	5449	San Agustín de Valle F
	Justinn	Reidie	diabetic	33 Morning Street Room 549	SLC	Santa Lucija
	Silvain	Kliche	gluten-free	27705 Kenwood Place 6th Floor	1617	Shuangyang
	Kimble	Dowthwaite	gluten-free	45 Sachtjen Hill PO Box 89766	2425	Cimanggu
	Zsa zsa	Keyser	gluten-free	64 Cardinal Road Suite 76	2829	Liujiachang
	Willyt	Gannon	gluten-free	93250 Hauk Park PO Box 40990	30220	Sam Roi Yot
	Brewer	Skellorne	gluten-free	709 Pankratz Court 3rd Floor	3435	Bakau
	Darell	Davenhall	gluten-free	0712 Monterey Trail Room 214	357367	Kabakovo
	Rowney	Pearsall	gluten-free	6738 Clove Road 17th Floor	4405	Rosario de Lerma
	Aveline	Risbrough	gluten-free	5 Pennsylvania Pass PO Box 47200	5678	Yong'an
	Vito	Pontefract	gluten-free	00894 Buell Circle PO Box 16234	6311	Guatraché
	Ardelis	Weinberg	gluten-free	57 Thierer Road PO Box 62500	93344 ...	Le Raincy
	Tades	Thiem	vegan	6 Little Fleur Hill 6th Floor	1213	Maopingchang
	Arther	Negro	vegan	6 Northport Crossing Apt 55	1819	Pukou
	Elvira	Berardt	vegan	0620 Drewry Place Suite 33	3233	Dingchang
	Hayden	Knoller	vegan	5 Summit Trail Apt 1808	347252	Sokol
	Tessi	Trumpeter	vegan	65749 Sachtjen Avenue Apt 1655	64525-...	Várzea Grande
	Stillman	Nelthropp	vegan	3 Farragut Court Room 1705	739 44	Brušperk
	Den	Tavernor	vegan	73361 Vernon Place PO Box 98949	96980	Hidalgo
	Gran	Barr	vegetarian	9 Crescent Oaks Crossing 17th Fl...	2627	Cimuncang
	Augusto	Wannes	vegetarian	81 Fieldstone Junction Apt 1558	3031	Donan
	Jorge	Meletti	vegetarian	21364 Golden Leaf Place 10th Floor	3759	Dimovo
	Aggi	Cator	vegetarian	25 Charing Cross Avenue Apt 967	439 83	Lubenec
	Cecilius	Biggin	vegetarian	41 Vera Crossing PO Box 60739	4423	Chicoana
	Keenan	Town	vegetarian	34167 School Crossing Apt 1474	75220 ...	Paris 16
	Jewel	Burbury	vegetarian	80 Charing Cross Road 9th Floor	9708	Cotabato
	Tommie	Kender	vegetarian	1316 McCormick Drive 20th Floor	J6W	Mascouche

3. The senior center is scheduling a pet therapy session for the elderly on 2022-12-18, to ensure the safety and comfort of the participating elderly. There are several criteria that must be met:

- The residents with known allergies to dog dander and cat dander must be excluded to prevent potential allergic reactions.
- Residents over 80 years old and those who have an upcoming medical appointment in the next 7 days shall not participate.
- The health parameter of the residents should be stable (Heart Rate: 60 – 100 beats per minute; Systolic blood pressure < 140 mmHg; Diastolic blood pressure > 60 mmHg)

Generate a list of the residents that are eligible to participate based on their latest health records, sort by the resident's name, date of birth, phone number, email, and their care ID.

Query

```

• SELECT
    CONCAT(r.first_name, ' ', r.last_name) AS Name,
    r.date_of_birth,
    r.phone_number,
    r.email,
    r.resident_id
FROM
    residents r
LEFT JOIN
    allergies a ON a.resident_id = r.resident_id
    and (a.allergy_source = 'dog dander' or a.allergy_source = 'cat dander')
LEFT JOIN
    appointment ap on ap.resident_id = r.resident_id and ap.appointment_date
    BETWEEN '2022-12-18' AND DATE_ADD('2022-12-18', INTERVAL 7 DAY)
LEFT JOIN
    health_records hr ON hr.resident_id = r.resident_id
    -- Use the latest health record for each resident
    AND hr.health_records_id = (
        SELECT MAX(hr2.health_records_id)
        FROM health_records hr2
        WHERE hr2.resident_id = r.resident_id
    )
WHERE
    a.allergy_id IS NULL -- Make sure the resident is not allergic to pets
    AND ap.appointment_id IS NULL -- No appointments within 7 days of the event
    AND TIMESTAMPDIFF(YEAR, r.date_of_birth, '2022-12-18') <= 80 -- Check resident's age
    AND hr.heart_rate BETWEEN 60 AND 100 -- Check resident's heart rate
    AND hr.systolic_blood_pressure < 140 -- Check Residents Systolic blood pressure
    AND hr.diastolic_blood_pressure > 60 -- Check Residents Diastolic blood pressure
ORDER BY
    CONCAT(r.first_name, ' ', r.last_name), r.date_of_birth, r.phone_number, r.email, r.resident_id;

```

Tabular Output:

	Name	date_of_birth	phone_number	email	resident_id
	Arther Negro	1947-06-05	336-427-9606	anegroh@parallels.com	18
	Justinn Reidie	1946-09-19	327-906-1902	jreidiem@webmd.com	23
	Kimble Dowthwaite	1955-05-12	184-690-6459	kdowthwaiten@google.com	24
	Zsa zsa Keysel	1952-12-09	814-412-1927	zkeyselq@surveymonkey.com	27

4. A resident in the Center needs to be transferred to a hospital to start more comprehensive treatments. To ensure a smooth transition, a comprehensive health report of the resident needs to be generated for the hospital staff.

This procedure accepts a resident ID and will join the resident's info into five tables: Resident information (personal info, emergency contact, special care needs), Ailments (existing conditions and the resident's doctor info), Medications, Allergies, and past health records of past appointments.

Query:

```

1057 -- Stored procedure for generating a comprehensive health report for healthcare providers
1058 DELIMITER //
1059 • CREATE PROCEDURE GenerateHealthReport(IN residentID INT)
1060 BEGIN
1061     -- Resident's personal information including special care information
1062     SELECT
1063         'Resident Information' AS Section,
1064         r.First_Name, r.Last_Name, r.Date_of_Birth, r.Phone_Number, r.Email, r.Gender,
1065         CONCAT(e.First_Name, ' ', e.Last_Name) AS EmergencyContact,
1066         e.Phone_Number AS EmergencyContactPhone, e.Email AS EmergencyContactEmail,
1067         s.Dietary, s.Outdoor
1068     FROM Residents r
1069     LEFT JOIN Emergency_Contact e ON r.Emergency_Contact_ID = e.Emergency_Contact_ID
1070     LEFT JOIN Special_Care s ON r.Care_ID = s.Care_ID
1071     WHERE r.Resident_ID = residentID;
1072     -- Resident's ailments
1073     SELECT
1074         'Ailments' AS Section,
1075         a.Ailment_Name, a.Description,
1076         CONCAT(d.First_Name, ' ', d.Last_Name) AS Doctor,
1077         d.Specialization
1078     FROM Resident_Ailment ra
1079     JOIN Ailments a ON ra.Ailment_ID = a.Ailment_ID
1080     JOIN Doctors d ON ra.Doctor_ID = d.Doctor_ID
1081     WHERE ra.Resident_ID = residentID;
1082     -- Resident's medications
1083     SELECT
1084         'Medications' AS Section,
1085         m.Generic_Name, p.Dosage, p.Frequency, p.Start_Date, p.End_Date,
1086         CONCAT(d.First_Name, ' ', d.Last_Name) AS PrescribingDoctor
1087     FROM Prescriptions p
1088     JOIN Medications m ON p.Medication_ID = m.Medication_ID
1089     JOIN Doctors d ON p.Doctor_ID = d.Doctor_ID
1090     WHERE p.Resident_ID = residentID;
1091     -- Resident's allergies
1092     SELECT
1093         'Allergies' AS Section,
1094         a.Allergy_Type, a.Allergy_Source, a.Severity_Level, a.Checkup_Date
1095     FROM Allergies a
1096     WHERE a.Resident_ID = residentID;
1097     -- Resident's recent health records
1098     SELECT
1099         'Health Records' AS Section,
1100         h.Checkup_Date, h.Systolic_Blood_Pressure, h.Diastolic_Blood_Pressure,
1101         h.Heart_Rate, h.Weight, h.Height, h.Notes
1102     FROM Health_Records h
1103     WHERE h.Resident_ID = residentID
1104     ORDER BY h.Checkup_Date DESC
1105     LIMIT 5; -- Get the last 5 health records
1106 END //
1107 DELIMITER ;

```

Tabular output:

Section	First_Name	Last_Name	Date_of_Birth	Phone_Number	Email	Gender	EmergencyContact	EmergencyContactPhone	EmergencyContactEmail	Dietary	Outdoor
▶ Resident Information	Eibert	Twinterman	1932-05-13	950-157-3535	etwinterman1@hatena.ne.jp	M	Herta Gorner	932-165-5094	hgorner1@dailyml.co.uk	dairy-free	wheelchair access
Section	Aliment_Name	Description	Doctor		Specialization						
▶ Aliments	Stroke	Curabitur in libero ut massa volutpat convallis. ...	Gabriele Shaddick		Gastroenterology						
Section	Generic_Name	Dosage	Frequency	Start_Date	End_Date	PrescribingDoctor					
▶ Medications	Warfarin	4	twice a day	2020-01-26	2024-05-30	Kariotta Kann					
Section	Allergy_Type	Allergy_Source	Severity_Level	Checkup_Date							
▶ Allergies	Pet dander	Peanuts	8	2021-11-18							
Section	Checkup_Date	Systolic_Blood_Pressure	Diastolic_Blood_Pressure	Heart_Rate	Weight	Height	Notes				
▶ Health Records	2024-06-21	120	80	40	300	65	test insert hr2, low heart rate and abnormal BMI				
Health Records	2021-09-11	141	86	136	217	58					
Health Records	2020-11-29	131	89	101	178	65	Phasellus sit amet erat. Nulla tempus. Vivamus l...				

5. After a doctor finishes an appointment with a resident, the doctor will add a new health record for the patient. If the patient's status is abnormal, the care center needs to be alerted so that the staff can provide additional care.

This trigger will create alert whenever the new health record's blood pressure, heart rate, or BMI is abnormal.

Query:

```

1005 -- Alert when abnormal health stats detected during a checkup
1006 DELIMITER //
1007 • CREATE TRIGGER HealthCheckupAlert
1008 AFTER INSERT ON Health_Records
1009 FOR EACH ROW
1010 BEGIN
1011     DECLARE alertMessage TEXT;
1012     DECLARE residentName VARCHAR(101);
1013
1014     -- Fetch the resident's full name
1015     SELECT CONCAT(First_Name, ' ', Last_Name) INTO residentName
1016     FROM Residents
1017     WHERE Resident_ID = NEW.Resident_ID;
1018
1019     -- Check for high blood pressure
1020     IF NEW.Systolic_Blood_Pressure > 140 OR NEW.Diastolic_Blood_Pressure > 90 THEN
1021         SET alertMessage = CONCAT('High blood pressure detected for ', residentName, ' on ', NEW.Checkup_Date);
1022         INSERT INTO Alerts (Resident_ID, Alert_Message) VALUES (NEW.Resident_ID, alertMessage);
1023     END IF;
1024
1025     -- Check for low heart rate
1026     IF NEW.Heart_Rate < 60 THEN
1027         SET alertMessage = CONCAT('Low heart rate detected for ', residentName, ' on ', NEW.Checkup_Date);
1028         INSERT INTO Alerts (Resident_ID, Alert_Message) VALUES (NEW.Resident_ID, alertMessage);
1029     END IF;
1030
1031     -- Check for high BMI (using inches and pounds)
1032     IF (NEW.Weight * 703) / (NEW.Height * NEW.Height) > 30 THEN
1033         SET alertMessage = CONCAT('High BMI detected for ', residentName, ' on ', NEW.Checkup_Date);
1034         INSERT INTO Alerts (Resident_ID, Alert_Message) VALUES (NEW.Resident_ID, alertMessage);
1035     END IF;
1036 END //
1037 DELIMITER ;

```

Tabular output of alerts:

	Alert_ID	Resident_ID	Alert_Message
▶	1	1	Appointment with Travus Spring on 2024-06-21
	2	2	Appointment with Gabriele Shaddick on 2024-06-21
	3	1	High blood pressure detected for Tessi Trumpeter on 2024-06-21
	4	2	Low heart rate detected for Elbert Twinterman on 2024-06-21
	5	2	High BMI detected for Elbert Twinterman on 2024-06-21
*	NULL	NULL	NULL

6. The care center celebrates residents' birthdays regularly. To do that, the staff needs a list of upcoming birthdays in the upcoming month.

This procedure will filter all residents whose birthday will happen in the next month. It will also calculate the residents' current ages.

Query:

```

1039  -- Stored procedure for getting upcoming birthdays in the next month
1040  DELIMITER //
1041  • CREATE PROCEDURE GetUpcomingBirthdays()
1042  BEGIN
1043      SELECT
1044          First_Name,
1045          Last_Name,
1046          Date_of_Birth,
1047          Phone_Number,
1048          Email,
1049          TIMESTAMPDIFF(YEAR, Date_of_Birth, CURDATE()) AS Current_Age
1050      FROM
1051          Residents
1052      WHERE
1053          MONTH(Date_of_Birth) = MONTH(CURDATE() + INTERVAL 1 MONTH);
1054  END //
1055  DELIMITER ;

```

Tabular output:

	First_Name	Last_Name	Date_of_Birth	Phone_Number	Email	Current_Age
▶	Tessi	Trumpeter	1965-07-22	890-344-0670	ttrumpeter0@dailyemotion.com	58
	Den	Tavernor	1968-07-02	672-888-9090	dtavernor2@netvibes.com	55
	Aveline	Risbrough	1957-07-26	589-743-0468	arisbrough4@delicious.com	66
	Cecilus	Biggin	1948-07-24	804-516-9086	cbiggin9@360.cn	75

Conclusions

The Senior Center Care Database provides a robust framework for managing the critical needs of a senior residence/care center. The strength of the database application lies in the

comprehensiveness of the resident and health information, and the related alerts for the staff. This tool would allow for effective appointment scheduling, and thorough care coordination between the center, the doctors and the family of the residents. The database is designed to streamline the care and logistics for residents, and handle various aspects of residents' care, including personal information, medical details, appointments, prescriptions, caregivers, health records, and emergency events – which are critical to the immediate as well as long-term health and care of the residents.

This project establishes a strong foundation for a database application, providing comprehensive functionality. While it currently lacks a user interface and advanced encryption/security measures, these elements were outside the scope of the project and can be integrated in future iterations. To demonstrate the database's capabilities, we utilized mock data from an online generator due to the confidentiality of real health data. Although this mock data is not perfectly realistic, it sufficiently illustrates the functionality of the system.

This project has successfully demonstrated a comprehensive, working back-end database, and with additional front-end integrations, the application could be fully functional and scalable for businesses.

Author Contributions

The team collaborated effectively, with each member contributing uniquely to the project's successful completion. Although everyone was involved in all aspects, the level of contribution varied for each part. Bingchun focused on data cleaning, data import, additional database design, ER diagram edits, and the report. Rediet worked on the initial database design, ER diagram, report, and presentation. Weiwei contributed to the idea conception, additional database design, ER diagram edits, presentation, and report. Yifei handled additional database design, procedures and triggers, use cases, and the presentation.