
Home Care Database Management System

Submitted by :-

Mitul dudhat(202318024)

Harshil shah(202318033)

Guided by Prof. Minal Bhise



**DA-IICT, Near Indroda Circle,
Gandhinagar 382 007, Gujarat (India)**

November, 2022

Content

1. SRS	1
1. Case Study	
1.1 Purpose	2
1.2 Intended Audience	4
1.3 Product Scope	5
1.4 Description	6
2. Requirement Collection	
2.1 Background Reading	12
2.2 Interview	13
2.3 Questionnaires	15
2.4 Observation	28
2.5 Fact Finding Chart	32
3. List of Requirement	34
4. User Categories & Privileges	
4.1 List of Users	36
4.2 Privileges of Users	38
5. List of Assumptions	40
6. Business Constraints	42
2. ERD	44
1. Noun Analysis Table	45
2. Entity-Attribute Table	58
3. ERD	59
3. Database Schema	
1. Relational Schema	63
2. Schema Refinement Process	64
3. Normalized Schema	70
4. DDL and SQL	
1. DDL Script	74
2. Plain Text and SQL Queries	94
3. Executed Queries Outputs	96

ACKNOWLEDGEMENT

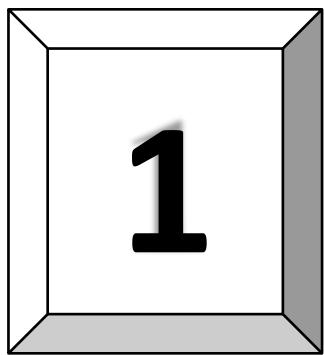
We would like to give our sincere acknowledgement to everybody responsible for the successful completion of our project “Home Care Database Management System”.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of this project.

We owe our deepest gratitude to our project guide **Ta.Kalgi Gandhi and Ta.Vedant dave**, who took interest on our project work and guided us all along till the completion of our project work by providing all the necessary help for developing a good Database System. We would also like to admiringly acknowledge Prof. **Minal Bhise** for moral support which helped us to accomplish our goal.

We would also like to thank all our lecturers.

Last but not the least we are also thankful to all friends & those who have contributed directly or indirectly by giving their suggestions or even a advice for the completion of this project.



Section : SRS Document

Home Care Database Management System

1. Case Study

→Introduction

In an era marked by a growing aging population and an increasing demand for personalized healthcare, the role of home care services has become pivotal in providing essential support to individuals in need. Home care agencies face the formidable challenge of delivering high-quality care while effectively managing the complex web of client information, caregiver schedules, billing processes, and regulatory compliance. To meet these challenges head-on and to enhance the overall home care experience for clients, caregivers, and agency staff, the Home Care Worker Database Management System has been developed

- **In Home care worker we have mainly Six type of worker Associative care provider :**

1. Assistive care providers
2. Personal care aides
3. Home health aides
4. Certified nursing assistants
5. Skilled care providers
6. Therapists

1.1 Purpose

The purpose of the Home Care Worker Providing Agency Database Management System is diverse, involving various important goals and advantages that cater to the requirements of home care agencies, clients, caregivers, and regulatory bodies. Here's a summary of the main aims of this system:

1. Efficient Client and Caregiver Management:

The system serves as a centralized repository for storing and managing detailed information about clients and caregivers. This includes demographic data, medical histories, care plans, availability, and preferences.

Purpose: To streamline the process of matching clients with suitable caregivers, ensuring personalized care and efficient scheduling.

2. Optimized Caregiver Scheduling:

Automated scheduling tools within the system help agencies match client needs with caregiver availability, minimizing scheduling conflicts and optimizing resource allocation.

Purpose: To enhance the efficiency of caregiver scheduling, reducing missed appointments and ensuring timely care provision.

3. Billing and Financial Management:

The system generates invoices, tracks payments, and provides financial insights, simplifying billing processes for both agencies and clients.

Purpose: To facilitate accurate and transparent financial transactions, improving financial management and reducing billing disputes.

4. Performance Monitoring:

Integrated metrics and feedback mechanisms allow agencies to monitor caregiver performance, track key performance indicators, and identify areas for improvement.

Purpose: To enhance the quality of care by enabling agencies to assess and improve caregiver performance.

5. Robust Reporting and Analytics:

The system offers reporting and analytics tools to generate detailed insights into agency performance, client satisfaction, and financial metrics.

Purpose: To support data-driven decision-making, enabling agencies to make informed choices and continually enhance their services.

6. Integration with Existing Systems:

Seamless integration capabilities enable the system to work alongside existing software solutions, such as accounting systems and scheduling tools.

Purpose: To minimize data duplication, improve operational efficiency, and ensure compatibility with an agency's existing technology stack.

7. Data Security and Access Control:

Stringent security measures, including data encryption and role-based access control, safeguard sensitive client and caregiver information.

Purpose: To protect client confidentiality, comply with data protection regulations, and maintain the integrity of the database.

8. Scalability:

The system is designed to accommodate agency growth, handling an increasing number of clients, caregivers, and data without sacrificing performance.

Purpose: To support agencies as they expand their operations and scale their services to meet growing demand.

9. Regulatory Compliance Support:

The system assists agencies in maintaining compliance with healthcare regulations by automating record-keeping, reporting, and auditing processes.

Purpose: To reduce administrative burdens and ensure that agencies adhere to regulatory standards.

10. Enhanced Communication:

The system includes communication features that facilitate real-time interaction between clients, caregivers, and agency staff, improving coordination and client satisfaction.

Purpose: To enable effective and transparent communication, enhancing the overall care experience.

In summary,

The Home Care Worker Providing Agency Database Management System serves the primary purpose of optimizing home care agency operations, improving the quality of care delivered to clients, and ensuring compliance with regulatory requirements. It achieves these objectives by offering a comprehensive set of features that streamline administrative tasks, enhance caregiver-client coordination, and provide valuable data insights for informed decision-making. Ultimately, the system's purpose is to elevate the standard of home care services, benefiting all stakeholders involved in the care delivery process.

1.2 Target Audience

The targeted audience for the Home Care Worker Agency (HCWA) system typically includes individuals and entities involved in the management, operation, and utilization of the system within a home care agency. Here are the primary targeted audiences for the HCWA system:

Home Care Agency Owners/Management: The leadership team of the home care agency is a key audience. They are responsible for the overall strategic direction, decision-making, and implementation of the HCWA system within the organization.

Administrative Staff: Administrative personnel within the home care agency, such as receptionists, billing specialists, and administrative assistants, are important users of the system. They use it for tasks related to client management, billing, record-keeping, and scheduling.

Home Care Workers (Caregivers): Caregivers who provide direct care to clients are a vital audience. They use the system to access schedules, record client information, submit timesheets, and communicate with the agency.

Clients and Their Families: The clients receiving home care services and their families or caregivers (if applicable) may also interact with the system. They might use it to view schedules, provide feedback, and access billing and care information.

1.3 Product Scope

Certainly, the product scope defines the boundaries of the software system – what it will and will not include. In the context of the Home Care Worker Providing Agency Database Management System, the product scope is as follows

1. Home Care Worker Profiles:

The system will allow administrators to create, edit, and deactivate home care worker profiles. These profiles will include personal information, skills, certifications, and availability.

2. Client Profiles:

Administrators can manage client profiles, including personal information, medical needs, and preferences. Clients will be able to view their assigned home care worker's information.

3. Scheduling and Assignment:

The system will enable Home Care Coordinators to assign home care workers to clients based on availability and skills. Home care workers can view their schedules and assigned tasks. Automatic notifications will be sent for new assignments and schedule changes.

4. Service Records:

Home Care Workers can record services provided, including date, time, and service details. Clients will have access to a history of services provided.

5. Reporting:

The system will allow administrators and coordinators to generate reports on service utilization, worker assignments, and client feedback. These reports can be exported in various formats.

1.4 Description

Key Features:

- Comprehensive Client Profiles:

The system empowers home care agencies to create and manage comprehensive client profiles, encompassing demographics, medical histories, care plans, and preferences. This holistic view of the client allows for personalized and effective care delivery.

- Efficient Caregiver Scheduling:

Automation takes center stage as the system optimizes caregiver scheduling. It matches client needs with caregiver availability, ensuring timely care provision while minimizing scheduling conflicts.

- Billing and Financial Management:

The system streamlines financial operations by generating invoices, tracking payments, and providing financial insights. This simplifies billing for both agencies and clients.

- Performance Monitoring:

Caregiver performance evaluation is simplified through integrated metrics and feedback mechanisms. Agencies can track and improve caregiver performance, enhancing overall service quality.

- Security and Access Control:

Stringent security measures, including role-based access control and data encryption, safeguard sensitive client and caregiver information, ensuring compliance with data protection regulations.

- User-Friendly Interface:

The system boasts an intuitive, user-friendly interface, requiring minimal training. Users can easily access vital information and perform tasks efficiently.

- Scalability:

Designed for scalability, the system accommodates the agency's growth, handling an increasing number of clients, caregivers, and data without compromising performance.

Benefits

- Enhanced Care Quality:

The Home Care Database Management System promotes the highest standards of care by facilitating efficient scheduling, performance monitoring, and client feedback collection.

- Operational Efficiency:

Streamlined administrative tasks, reduced manual effort, and improved resource allocation translate to operational excellence.

- Data-Driven Decision-Making:

Data analytics empower agencies to make informed decisions, enabling optimization of agency performance.

- Improved Client Satisfaction:

Timely and personalized care, based on detailed client profiles, leads to higher client satisfaction rates.

Conclusion:

The Home Care Database Management System represents a critical tool for modern home care agencies, offering exceptional care services, streamlined operations, and a competitive edge. In a rapidly evolving field where the demand for high-quality home care is growing, this system ensures that agencies are well-equipped to meet the diverse needs of their clients, uphold the highest standards of care, and provide an unparalleled level of service. Its impact extends beyond the operational realm, positively affecting the lives of those receiving care, their families, and the dedicated caregivers who serve them. In an era where the importance of home-based care is increasingly recognized, this system stands as an invaluable asset, driving the evolution of home care services toward excellence and personalization.

Number of Services Provided:

Our Home Care Database Management System offers a diverse array of services, encompassing a minimum of ten essential areas:

1. Medical Care:

Our system supports the management of medical care, including wound care, medication management, and medical monitoring, ensuring clients receive necessary medical attention.

2. Personal Care:

Caregivers can efficiently assist clients with activities of daily living (ADLs) such as bathing, dressing, grooming, toileting, and mobility support

3. Meal Preparation:

Our system facilitates meal planning and preparation, catering to clients' dietary needs and preferences to ensure proper nutrition.

4. Medication Reminders:

Caregivers can help clients manage medication schedules, ensuring medications are taken correctly and on time.

5. Light Housekeeping:

Our system assists with light housekeeping tasks, including cleaning, laundry, and organization, to maintain a safe and comfortable living environment.

6. Transportation:

We offer transportation services, helping clients with errands, medical appointments, and social outings.

7. Respite Care:

Our respite care service provides temporary relief to family caregivers, allowing them to take a break from caregiving responsibilities while ensuring their loved ones continue to receive care.

8. Specialized Care:

Our agency provides specialized care for clients with specific conditions such as Alzheimer's disease, Parkinson's disease, or diabetes. This involves tailored care plans and caregiver training.

9. Palliative and Hospice Care:

We offer palliative and hospice care services to provide comfort, pain management, and emotional support to individuals with life-limiting illnesses and their families.

The variety and breadth of services provided by our Home Care Database Management System empower home care agencies to address the diverse needs of their clients comprehensively, ensuring that clients receive personalized, high-quality care that fosters comfort, independence, and well-being

Challenges in Home Care Service Delivery:

Home care services play a vital role in providing essential healthcare and support to individuals in the comfort of their homes. However, this unique mode of care delivery presents several challenges that home care agencies must address to ensure the well-being and satisfaction of both clients and caregivers.

- Coordination of Care:

One of the foremost challenges in home care service delivery is the coordination of care services. Clients often require a range of services from multiple caregivers and healthcare professionals. Coordinating these services to ensure seamless and efficient care can be complex. For example, a client may need assistance with medication management, physical therapy, and personal care, each provided by different caregivers and specialists. Without effective coordination, there is a risk of overlap, missed appointments, and inefficiencies in care delivery.

- Privacy Concerns:

Privacy and data security are paramount in the home care setting. Clients trust agencies with their sensitive medical information and personal details. Ensuring the confidentiality of this information is not only a legal requirement but also a fundamental ethical obligation. The challenge lies in implementing robust data protection measures to safeguard client data, including electronic health records, while still allowing authorized caregivers and agency staff access to the necessary information for providing care.

- Caregiver Matching:

Matching the right caregiver with the right client is crucial for a positive care experience. Every client has unique needs, preferences, and personalities, and not all caregivers are equally suited to every client. Challenges arise when agencies must find caregivers with the specific skills, training, and personality traits that align with a client's requirements. A mismatch can lead to client dissatisfaction, decreased quality of care, and increased caregiver turnover.

- Cost Management:

Understanding the costs associated with home care services can be challenging for clients and their families. Clients often need to navigate complex insurance policies, out-of-pocket expenses, and varying rates for different services. Without clear guidance and support, clients may struggle to manage these costs, leading to financial stress and uncertainty.

- Client Communication:

Effective communication between clients, caregivers, and agency staff is crucial for delivering high-quality care. Challenges can arise when there are breakdowns in communication, such as

missed messages or misunderstandings about care plans and schedules. Inadequate communication can lead to errors in care delivery and decreased client satisfaction.

- Emergency Response:

Quick and efficient response to emergencies and unexpected situations is critical for client safety. Home care clients may face sudden medical issues or accidents that require immediate attention. Ensuring that caregivers are well-prepared to respond to emergencies and that agencies have protocols in place for timely assistance can be a significant challenge.

- Regulatory Compliance:

The healthcare industry is subject to constantly evolving regulations and standards. Home care agencies must navigate a complex web of federal, state, and local regulations to ensure compliance. Staying up-to-date with changing requirements and maintaining meticulous records to demonstrate compliance can be burdensome for agencies.

- Data Management:

Home care agencies must manage a vast amount of data, including client records, caregiver profiles, schedules, and billing information. Maintaining accurate records while safeguarding sensitive client data is a considerable challenge. Data breaches or inaccuracies in records can have serious consequences, including legal and financial repercussions.

Solutions to Address These Challenges:

Our Home Care Database Management System is specifically designed to address these challenges effectively and provide solutions that enhance the quality of care and streamline agency operations.

- Care Coordination:

Our system offers streamlined care coordination tools that allow agencies to manage complex care schedules effortlessly. It matches client needs with caregiver availability, ensuring that clients receive the right care at the right time. This minimizes scheduling conflicts, reduces missed appointments, and enhances the overall client experience.

- Privacy Protection:

We prioritize data security with robust measures, including encryption and role-based access control. Our system ensures that client information remains confidential while still allowing authorized users, such as caregivers and agency staff, to access the necessary data to provide

care. This not only meets regulatory requirements but also instills trust in clients and their families.

- Financial Guidance:

We provide financial guidance tools to assist clients in understanding and managing the costs of home care services. This includes clear explanations of insurance coverage, out-of-pocket expenses, and available payment options. Our aim is to alleviate financial stress and provide transparency in billing.

- Client Communication:

Our system facilitates effective communication between clients, caregivers, and agency staff. Features such as messaging, appointment reminders, and care plan updates ensure that everyone involved is well-informed, reducing the risk of misunderstandings and errors in care delivery.

- Emergency Response Protocols:

We have integrated emergency response protocols into our system to ensure quick and efficient reactions to critical situations. Caregivers have access to resources and guidance for responding to emergencies, promoting client safety and peace of mind.

- Regulatory Compliance Support:

Our system simplifies regulatory compliance by automating record-keeping and reporting processes. Agencies can easily generate the necessary documentation to demonstrate compliance with healthcare regulations, reducing administrative burdens and minimizing compliance risks.

- Data Security:

Stringent data security measures, including encryption, access control, and regular data backups, protect sensitive information and maintain the integrity of client and agency data. Our commitment to data security ensures that agencies can trust our system with their most valuable asset—their data.

In conclusion, the challenges inherent in home care service delivery are multifaceted, but our Home Care Database Management System is designed to address these challenges head-on. We provide comprehensive solutions that enhance care coordination, protect client privacy, optimize caregiver matching, offer financial guidance, facilitate communication, ensure emergency preparedness, simplify regulatory compliance, and prioritize data security. With our system, home care agencies can overcome these challenges and provide the highest quality of care to their clients, ultimately improving the overall home care experience for both clients and caregivers.

2. Document the Requirements Collection/ Fact Finding Phase

→ **Document the input and the output for the requirements collection techniques used by you:**

2.1 Background Reading

During the requirements collection phase of the Home Care Worker Providing Agency Database Management System project, various techniques were employed to gather essential information and insights. These techniques included background reading and conducting interviews

Title: "Effective Home Care Management Strategies"

- 2 **Description:** This book explores best practices for managing home care services, including scheduling, caregiver-client matching, and ensuring quality care. It provides insights into the challenges faced by home care agencies and ways to optimize operations.
Title: "Web-Based Database Management Systems"
- 3 **Description:** This online article discusses the benefits and challenges of developing web-based database management systems. It highlights the importance of user-friendly interfaces, data security, and real-time updates for improved efficiency.

Video : <https://www.youtube.com/watch?v=lTv53IhSwKk>

- **Description :**
 - A home cleaner discusses changes in the industry, the complexity of personal care, and the strain it puts on carers and companies
 - Home care can be stressful, especially when dealing with dementia patients.
 - Real-time monitoring affects personal care service

Reference link :

- 4 <https://www.als.org/navigating-als/resources/fyi-different-types-home-care-workers>
- 5 <https://www.cms.gov/Medicare/Quality-Initiatives-Patient-Assessment-Instruments/HomeHealthQualityInits>
- 7 <https://www.reqview.com/doc/iso-iec-ieee-29148-srs-example/>

Combined Requirements from Background Reading/s:

- A need for tailored care plans that can be easily updated based on client needs.
- Efficient scheduling to match caregivers' skills and availability with clients' requirements.
- The importance of electronic visit verification to ensure accurate tracking of care provided.
- Regulatory compliance requirements for home care agencies.

2.2 Interview

Interviewee:

Keyur Patel [Manager of Company]

Key Points Gathered:

- Clients' care needs vary, and personalized care plans are crucial. Scheduling is a challenge; an automated system to match caregivers' skills with clients' needs is desired.
- Caregivers' certifications and training need to be tracked for compliance.
- Accurate billing and invoicing based on services provided are essential. Data security and privacy are paramount due to sensitive client information.

company name :

Livonta Home Care Worker Providing Agency Database Management

Interviewer: What are the main goals and objectives of your home care agency, and how will this database management system support these goals?

Interviewee: Our primary goals are to provide top-quality care, efficient scheduling, and accurate record-keeping. The system will centralize data, automate scheduling, and help us monitor care quality and agency performance.

Interviewer: Who will be the primary users of this system, and what specific roles and permissions do you envision for each user type?

Interviewee: The primary users are agency administrators and caregivers. Administrators will have full access, while caregivers will access their schedules and client information. Clients may view their profiles for communication.

Interviewer: What types of data do you need to store and manage within the system?

Interviewee: We need to manage client demographics, medical histories, caregiver profiles, schedules, care plans, billing, and caregiver evaluations.

Interviewer: How do you envision data entry into the system, and who will be responsible for updating and maintaining this information?

Interviewee: Caregivers and administrative staff will enter data. Data entry will occur during client onboarding, caregiver hiring, and daily scheduling. Maintenance will be the responsibility of our admin team.

Interviewer: What types of reports or analytics would you like to generate from the system?

Interviewee: We need reports for caregiver performance, client satisfaction, financial data, scheduling efficiency, and compliance with regulations.

Interviewer: Are there any existing systems or software that this database management system needs to integrate with?

Interviewee: Yes, we use QuickBooks for accounting and Google Calendar for scheduling. Integration is crucial to streamline our operations.

Interviewer: How will you ensure the security of sensitive data within the system?

Interviewee: We plan to implement role-based access control, data encryption, regular security audits, and employee training to safeguard data. Only authorized personnel will access confidential information.

Interviewer: What are your expectations regarding the user interface's design and usability?

Interviewee: We expect an intuitive and user-friendly interface that requires minimal training. Users should access information and functions easily.

Interviewer: Do you have expectations regarding the system's performance and scalability?

Interviewee: We expect good performance even during peak periods and scalability to accommodate agency growth.

Interviewer: Do you have a specific budget and timeline for this project?

Interviewee: Our budget is \$X, and we aim to complete the project within 6-9 months to align with our expansion plans.

Interviewer: Is there any other functionality or requirements you'd like to add?

Interviewee: One additional requirement is a client feedback feature to rate caregiver performance and satisfaction.

Interviewer: Thank you for your input. Is there anything else you'd like to add before we conclude?

Interviewee: I believe we've covered everything. We appreciate your assistance with this project and look forward to the SRS document based on this interview.

2.3 Questionnaire :

- **Prepare questionnaire/s:**

1. Personal Information (Optional):

- Name
- Email Address
- Phone Number: [Text Field]

2. How did you hear about our services?

- Website
- Social Media
- Referral
- Other (Please specify): [Text Field]

3. On a scale of 1 to 5, please rate your overall satisfaction with our services:

- (1) Very Dissatisfied
- (2) Dissatisfied
- (3) Neutral
- (4) Satisfied
- (5) Very Satisfied

4. What do you like most about our services?

[Text Area]

5. What areas do you think we can improve?

[Text Area]

6. Were our services able to meet your expectations?

- Yes
- No (If no, please provide details in the next question)

7. If our services did not meet your expectations, please provide more details:

[Text Area]

8. Do you have any suggestions for new features or improvements to our services?

[Text Area]

9. How often do you use our services?

- Daily
- Weekly
- Monthly
- Rarely
- Never

10. What platform or device do you use to access our services?

- Desktop Computer
- Laptop
- Smartphone
- Tablet
- Other (Please specify): [Text Field]

11. Would you recommend our services to others?

- Yes
- No
- Maybe

12. Do you have any additional comments or feedback you'd like to share?

[Text Area]

13. May we contact you for further clarification or to discuss your feedback in more detail?

- Yes
- No

14. Specific Questions:

[Service Feedback] -

a. What specific service are you providing feedback on? What do you like most about this service?

[Text Area]

b. [For Website/App Feedback] - Which platform (website/app) are you providing feedback on? - Is the website/app easy to navigate and user-friendly? Please provide details. - Are there any technical issues or bugs you've encountered?

[Text Area]

c. [For Customer Support Feedback] - Did you recently interact with our customer support team? If yes, please share your experience.

[Text Area]

d. How would you rate the responsiveness and helpfulness of our support team? (Scale of 1 to 5)

[Text Area]

15. Would you like to subscribe to our newsletter for updates and promotions?

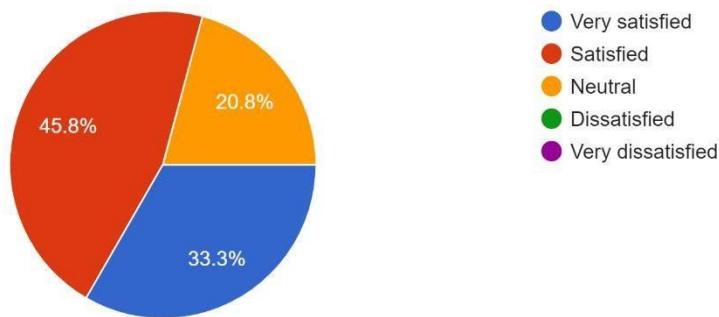
- Yes
- No

- **Summary**

Summary Clients Questionnaire

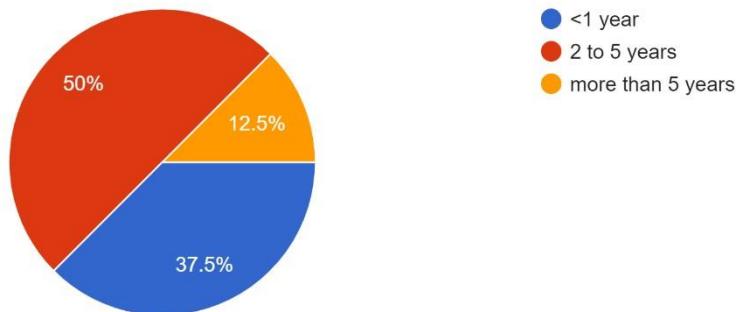
How satisfied are you with the current process of requesting home care services?

24 responses



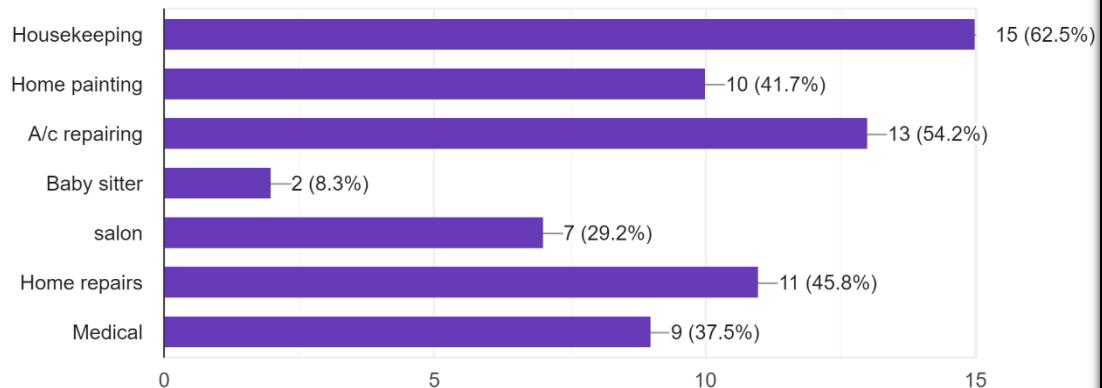
Required Experience Of Service Provider ?

24 responses



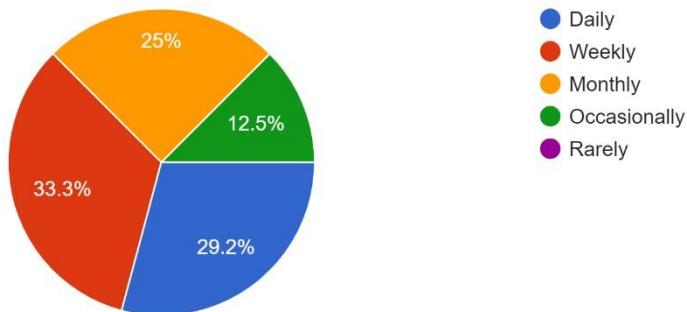
As a client Which service you use most?

24 responses



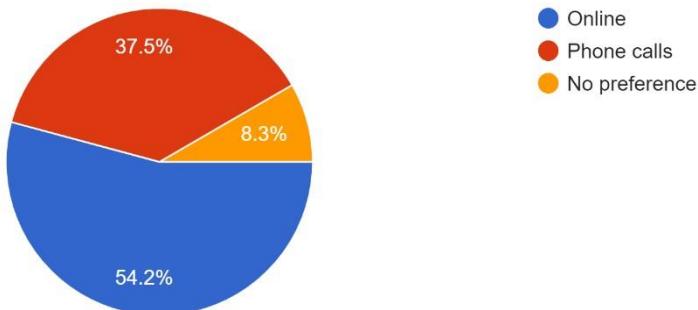
How often do you require home care services?

24 responses



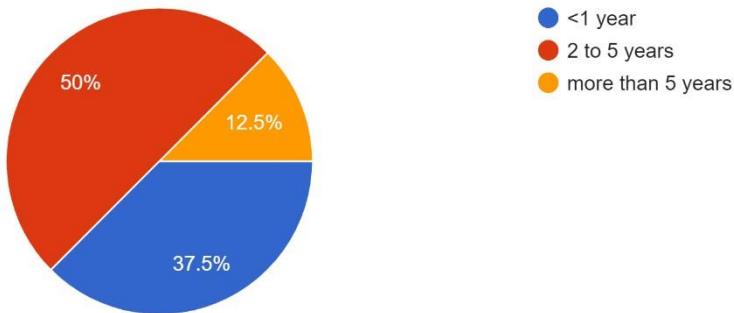
Would you rather receive appointment schedules and updates using a mobile app or other digital methods?

24 responses



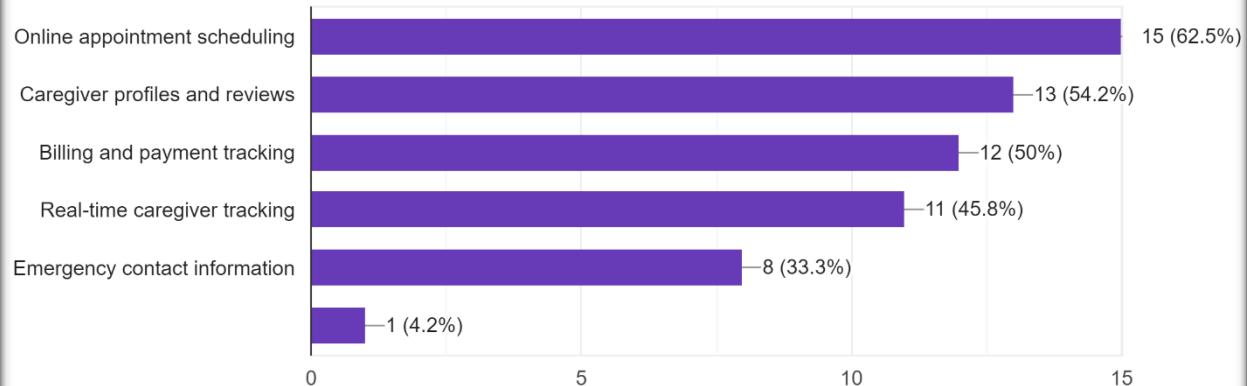
Required Experience Of Service Provider ?

24 responses



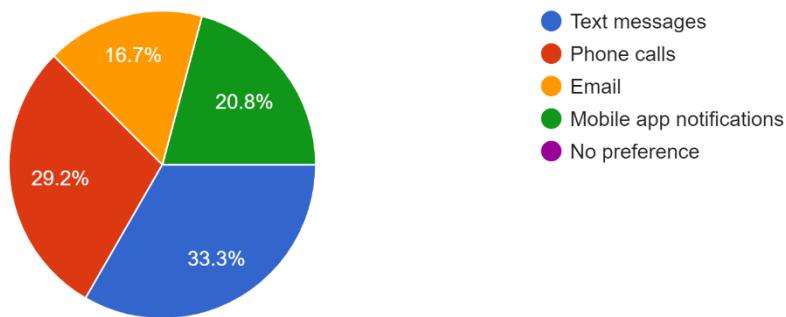
What features would you like in a home care service database? (Select all that apply)

24 responses



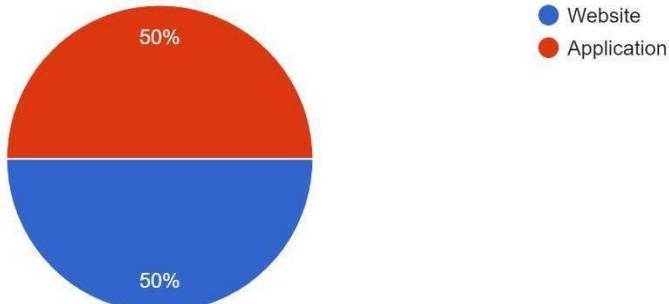
Which method of communication do you prefer for receiving appointment reminders and updates?

24 responses



Preferred Platform For system

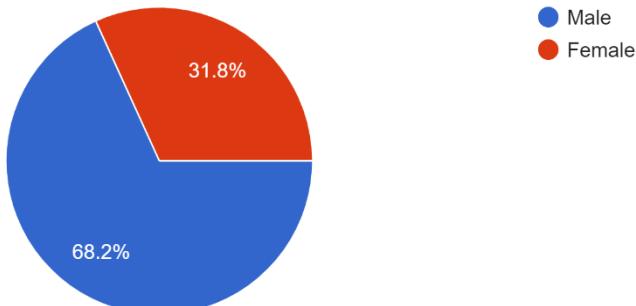
24 responses



Summary Caregivers Questionnaire

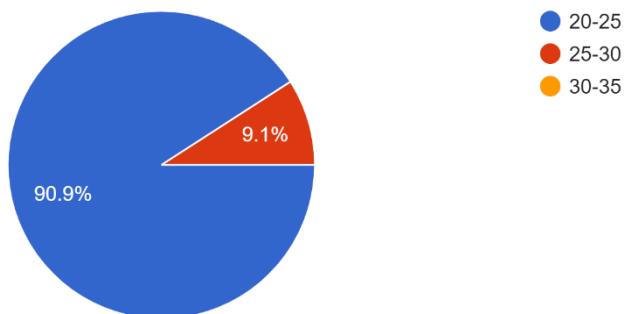
Gender:-

22 responses



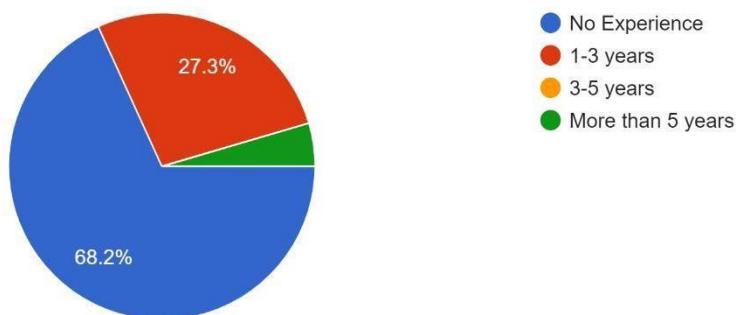
Age:-

22 responses



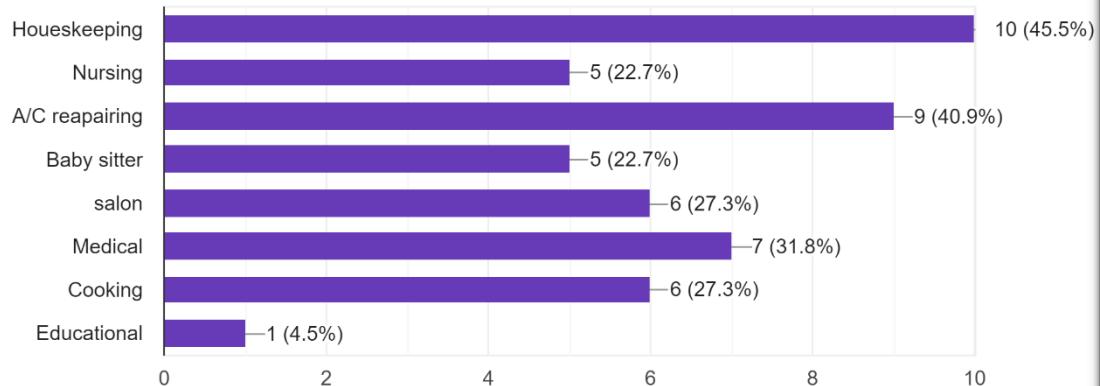
Work Experience

22 responses



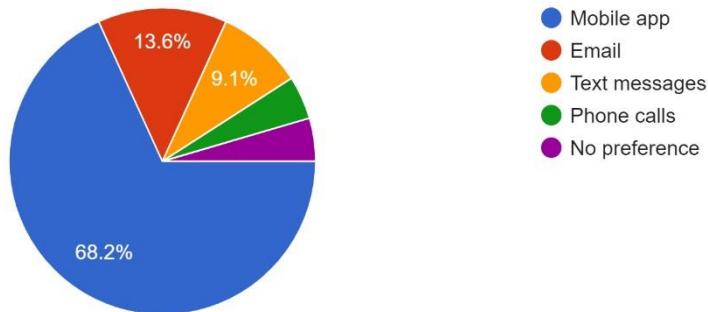
Which type of service you provide?

22 responses



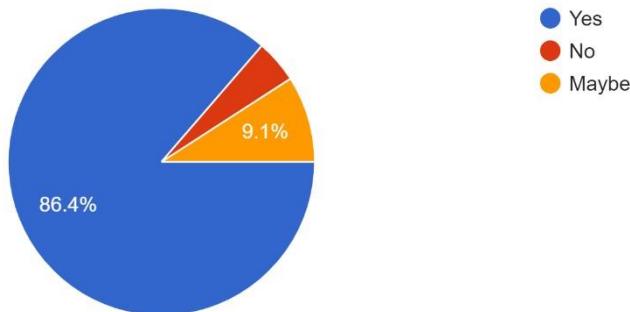
Would you rather receive appointment schedules and updates using a mobile app or other digital methods?

22 responses



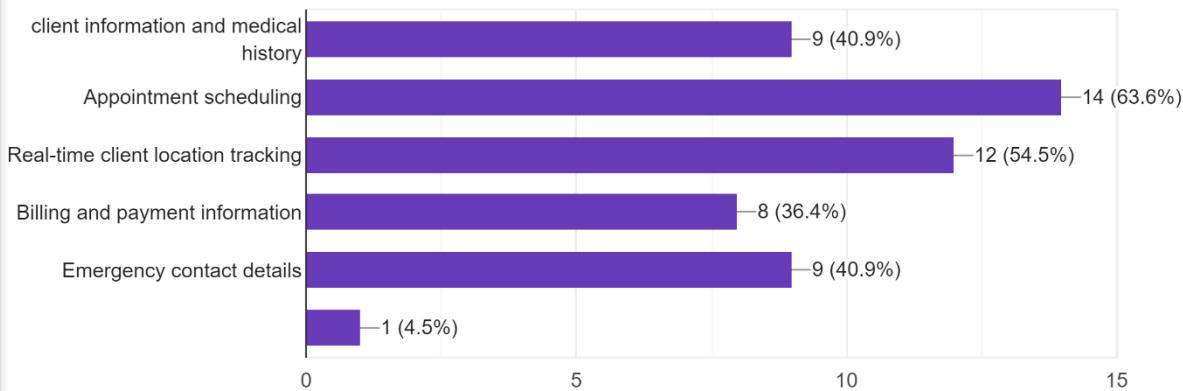
Would you find it helpful to have a feature that allows you to communicate with clients or their families through the database?

22 responses



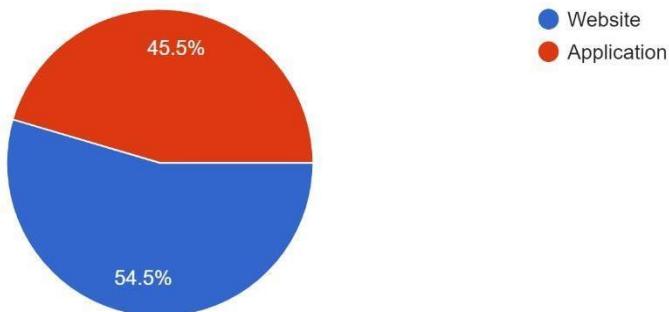
What features would you like in a home care service database for better job performance? (Select all that apply)

22 responses



Preferred Platform For system

22 responses



- **List the combined Requirements gathered from Response/s.**

1. Ability to maintain a database of client information, including personal details, medical history, care preferences, and contact information.
2. Capability to manage care workers, including their qualifications, schedules, and performance records.
3. Functionality to schedule care worker visits and assign them to clients based on availability and client needs.
4. Ability to record and track the services provided during care worker visits, including tasks performed and medications administered.
5. Features for generating invoices, tracking payments, and managing financial aspects of care services.
6. Tools to ensure compliance with regulatory guidelines and the ability to generate compliance reports.
7. Mobile applications or platforms to enable care workers to access and update client information in real-time.
8. Capability to scale the system to accommodate a growing number of clients, care workers, and services.

2.4 Observation/s

- **Write summary of your observation/s relevant to your database design.**

Based on client survey responses regarding home care services, several key summary points can be is :

1. **Satisfaction Levels:** Overall, clients have varying levels of satisfaction with the current process of requesting home care services. Responses range from "Very satisfied" to "Neutral," indicating that there is room for improvement in certain areas of the service.
2. **Service Usage:** Clients use a diverse range of home care services, including housekeeping, A/C repairing, home painting, and medical services. This diversity highlights the need for a comprehensive range of services to cater to various client needs.
3. **Digital Communication:** Many clients prefer receiving appointment schedules and updates through digital methods, such as mobile apps or online platforms. This suggests an increasing acceptance of technology for managing home care services.
4. **Experience Preferences:** Clients have different expectations regarding the experience level of service providers. Some clients prefer more experienced providers, while others are open to less experienced ones.
5. **Desired Features:** Clients express a variety of feature preferences for a home care service database. Commonly desired features include online appointment scheduling, caregiver profiles and reviews, billing and payment tracking, and real-time caregiver tracking.
6. **Communication Methods:** The preferred method of communication for appointment reminders and updates varies among clients. Text messages, email, and phone calls are all favored communication channels.
7. **Platform Preferences:** Clients have varying preferences for accessing the home care system, with some favoring mobile applications and others preferring websites. Offering both options may cater to a broader audience.
8. **Additional Feedback:** Some clients provide additional comments and feedback, offering valuable insights and suggestions for improving home care services.
9. **Overall Satisfaction:** While satisfaction levels vary, many clients express satisfaction with the services they receive. High satisfaction levels are a positive indicator for the home care agency.
10. **Frequency of Service Needs:** Clients' needs for home care services vary in frequency, with responses ranging from daily to monthly. This information can help agencies plan and schedule services efficiently.

11. Punctuality Concerns: One client mentioned the importance of appropriate timings, highlighting the significance of punctuality and scheduling accuracy in home care services.

From the provided "Caregivers Questionnaire", we can derive the following summary:

1. **Gender Distribution:** The respondents in the caregiver questionnaire are predominantly male, with most caregivers identifying as male.
2. **Age Distribution:** The majority of caregivers fall within the age group of 20-25 years, with a few in the 25-30 age range.
3. **Work Experience:** Many caregivers have little to no prior experience in the field, while some have 1-3 years of experience, and a few have more than 5 years of experience.
4. **Services Provided:** Caregivers offer a range of services, including nursing, baby-sitting, housekeeping, A/C repairing, medical, cooking, and salon services.
5. **Preferred Communication:** Caregivers predominantly prefer mobile apps for receiving appointment schedules and updates.
6. **Communication with Clients:** Most caregivers find it helpful to have a feature that allows them to communicate with clients or their families through the database.
7. **Desired Database Features:** Caregivers are interested in features like client information and medical history, appointment scheduling, real-time client location tracking, billing and payment information, and emergency contact details in a home care service database.
8. **Preferred Platform:** Caregivers express a preference for mobile apps and websites as their preferred platforms for using the system.
9. **Suggestions:** Caregivers provide various suggestions and feedback, including improvements related to appointment scheduling, communication, and negotiation processes.

- **List the combined Requirements gathered from Observation/s.**

Based on client survey responses regarding home care services, here is a list of combined requirements gathered:

1. Service Satisfaction Tracking:

- Collect feedback on the satisfaction level of clients with the home care service.

2. Frequency of Home Care Service:

- Understand how often clients require home care services (e.g., daily, weekly, monthly).

3. Preferred Home Care Services:

- Identify the types of services clients use the most (e.g., housekeeping, A/C repairing, medical).

4. Appointment Scheduling and Updates:

- Determine the preferred method for clients to receive appointment schedules and updates (e.g., mobile app, phone calls, text messages).

5. Service Provider Experience:

- Assess the required experience level of service providers from the client's perspective (e.g., 1-3 years, 2-5 years).

6. Database Features:

- Understand the desired features in a home care service database, such as billing and payment tracking, caregiver profiles and reviews, real-time caregiver tracking, emergency contact information.

7. Communication Preferences:

- Identify the preferred method of communication for receiving appointment reminders and updates (e.g., phone calls, text messages, email).

8. Preferred Platform for System:

- Determine whether clients prefer using a mobile application or a website for accessing the home care service.

9. Additional Comments and Feedback:

- Gather any additional comments or feedback from clients regarding their experience with home care services.

Based on Caregivers Questionnaire survey responses regarding home care services, here is a combined list of requirements gathered:

1. Appointment Scheduling:

- Caregivers prefer appointment scheduling features.
- Clients prefer online appointment scheduling.

2. Real-time Client Location Tracking:

- Caregivers find real-time client location tracking helpful.
- Clients are open to real-time caregiver tracking.

3. Communication Features:

- Caregivers are interested in features that allow them to communicate with clients or their families through the database.
- Clients prefer receiving appointment reminders and updates through phone calls, text messages, or mobile app notifications.

4. Client Information and Medical History:

- Caregivers are interested in accessing client information and medical history.
- Clients are open to caregivers having access to their medical history.

5. Billing and Payment Tracking:

- Caregivers want features related to billing and payment information.
- Clients are open to billing and payment tracking.

6. Emergency Contact Information:

- Caregivers and clients are interested in having access to emergency contact details.

7. Experience and Specialization:

- Caregivers have various levels of experience and specialize in different services (e.g., nursing, baby-sitting, medical, housekeeping, etc.).

8. Preferred Platform:

- Caregivers prefer mobile apps or other digital methods for receiving appointment schedules and updates.
- Clients have varying preferences for communication methods and platforms.

9. Additional Comments:

- Some caregivers and clients provided additional comments and feedback related to their preferences and experiences.

2.5 Create Fact Finding Chart :

Objective	Technique	Time Commitment
To understand the software design process	Background Reading	1-2 hours
To understand the healthcare industry	Background Reading	1-2 hours
To understand the nursing and personal care business	Background Reading	1-2 hours
Understanding how a small-scale DB is created and maintained	Case Study	1-2 days
To find how a home care service providing agency works	Interview (role playing)	0.5 day
To comprehend challenges faced by agency owner	Interview (role playing)	0.5 day
Identify areas for improvements and enhancements in the industry	Interview/Brainstorming	0.5 day

Gain insights into customers' desires and preferences	Questionnaire	30 minutes
Gain insights into service providers' desires and preferences	Questionnaire	30 minutes
To establish what kind of records and attributes should be kept in DB	Questionnaire Analysis	3 hours
To understand the current working environment and business constraints	Interview/Brainstorming	1 day
To understand software & hardware requirements	Case Study/Interview	6 hours

3. List of Requirements

→ Combine all the Requirements gathered

1. User Registration and Profiles:

- Users should be able to register as caregivers or clients.
- User profiles should include personal information, contact details, and work experience for caregivers.

2. Service Types:

- The system should support various types of home care services, such as nursing, babysitting, medical assistance, housekeeping, A/C repairing, cooking, and more.

3. Appointment Scheduling:

- Caregivers and clients should be able to schedule appointments.
- Real-time client location tracking should be available to caregivers.

4. Communication:

- The system should enable communication between caregivers, clients, and their families.
- Appointment reminders and updates should be sent via mobile app notifications, text messages, or phone calls based on user preferences.

5. Client Information:

- Caregivers need access to client information and medical history.
- Billing and payment tracking should be available for services rendered.

6. Caregiver Profiles and Reviews:

- Clients should be able to view caregiver profiles and read reviews.
- This information helps clients make informed decisions when choosing a caregiver.

7. Emergency Contact Information:

- The system should store emergency contact details for clients' safety.

8. Preferred Platform:

- Users have different platform preferences for accessing the system, including mobile apps, websites, text messages, and email.

9. Experience Level:

- Caregivers' experience levels vary, from less than one year to more than five years.

10. Additional Comments and Feedback:

- Users may provide feedback and comments for system improvement.

11. Client Satisfaction:

- Regularly gauge client satisfaction with home care services.

12. Frequency of Service Request:

- Understand how often clients require home care services.

13. Billing and Payment Tracking:

- Implement billing and payment tracking for transparency.

14. Negotiations and Timings:

- Ensure proper negotiations and appropriate timings for services.

15. Small-Scale Database Management:

- Understand how small-scale databases are created and maintained.

→ Remove duplicate requirements

After reviewing the combined list of requirements, here's a refined list with duplicate requirements removed:

1. User Registration and Profiles:

- Users should be able to register as caregivers or clients.
- User profiles should include personal information, contact details, and work experience for caregivers.

2. Service Types:

- The system should support various types of home care services, such as nursing, babysitting, medical assistance, housekeeping, A/C repairing, cooking, and more.

3. Appointment Scheduling:

- Caregivers and clients should be able to schedule appointments.
- Real-time client location tracking should be available to caregivers.

4. Communication:

- The system should enable communication between caregivers, clients, and their families.
- Appointment reminders and updates should be sent via mobile app notifications, text messages, or phone calls based on user preferences.

5. Client Information:

- Caregivers need access to client information and medical history.
- Billing and payment tracking should be available for services rendered.

6. Caregiver Profiles and Reviews:

- Clients should be able to view caregiver profiles and read reviews.
- This information helps clients make informed decisions when choosing a caregiver.

7. Emergency Contact Information:

- The system should store emergency contact details for clients' safety.

8. Preferred Platform:

- Users have different platform preferences for accessing the system, including mobile apps, websites, text messages, and email.

9. Experience Level:

- Caregivers' experience levels vary, from less than one year to more than five years.

10. Additional Comments and Feedback:

- Users may provide feedback and comments for system improvement.

11. Client Satisfaction:

- Regularly gauge client satisfaction with home care services.

12. Frequency of Service Request:

- Understand how often clients require home care services.

13. Billing and Payment Tracking:

- Implement billing and payment tracking for transparency.

14. Negotiations and Timings:

- Ensure proper negotiations and appropriate timings for services.

15. Small-Scale Database Management:

- Understand how small-scale databases are created and maintained.

4. User categories and privileges

4.1 List User Names with basic description of user role in the system/database

1. Administrator

- Description: Administrators have full access to the system and can manage user accounts, configure system settings, and oversee all aspects of the database. They often handle tasks like adding or removing users, setting access permissions, and ensuring data integrity.

2. Caregiver

- Description: Caregivers are frontline workers who provide direct care to clients in their homes. They use the system to access client information, update care plans, record vital signs, and report on the client's condition. Caregivers may also log their work hours and submit notes on the care provided.

3. Client

- Description: Clients are the individuals receiving care services. They may have limited access to the system to view their care schedules, communicate with caregivers or the agency, and provide feedback on the care they receive.

4. Scheduler/Coordinator

- Description: Schedulers or coordinators are responsible for assigning caregivers to clients, creating care schedules, and managing caregiver work shifts. They use the system to match caregiver skills with client needs and ensure proper coverage.

5. Billing and Finance Manager

- Description: This user role handles financial aspects of the home care services. They use the system to generate invoices, track payments, manage billing cycles, and reconcile financial records.

6. Training and Compliance Officer

- Description: Individuals in this role oversee caregiver training and ensure compliance with regulatory requirements. They may use the system to schedule training sessions, track caregiver certifications, and maintain compliance records.

7. HR Manager

- Description: Human Resources managers handle personnel-related tasks. In the database system, they can manage caregiver recruitment, employment records, performance evaluations, and handle HR-related documentation.

8. Quality Assurance Inspector

- Description: Quality assurance inspectors monitor and assess the quality of care provided by caregivers. They use the system to conduct inspections, review caregiver documentation, and ensure that care services meet established standards.

9. IT Support

- Description: IT support personnel provide technical assistance, troubleshoot system issues, and ensure the system's reliability and security.

10. Reports Analyst

- Description: Reports analysts extract and analyze data from the database to generate reports and insights for management. They help in decision-making by presenting data on caregiver performance, client satisfaction, and operational efficiency.

4.2 List privileges for each user categories

Administrator:

1. Create, modify, and delete user accounts.
2. Configure system settings, including security settings and access permissions.
3. Access and modify all client and caregiver records.
4. Generate system-wide reports and analytics.
5. Perform backups and data maintenance tasks.
6. Manage billing and financial records.
7. Assign and manage roles and privileges for other users.

Caregiver:

1. Access and update client information and care plans.
2. Record daily care activities and observations.

3. View their own work schedules and assignments.
4. Communicate with clients and coordinators through the system.
5. Submit time and attendance records.
6. Access training materials and complete required courses.
7. View client feedback and assessments relevant to their assigned clients.

Client:

1. View their own care schedule and caregiver assignments.
2. Provide feedback on the care they receive.
3. Communicate with their assigned caregiver or coordinator through the system.
4. Access relevant documentation about their care and services.
5. View and approve invoices and billing statements.

Scheduler/Coordinator:

1. Create and manage caregiver work schedules.
2. Assign caregivers to clients based on their skills and availability.
3. Access client and caregiver records for scheduling purposes.
4. Communicate with clients and caregivers through the system.
5. Generate reports related to caregiver assignments and schedules.
6. Monitor caregiver compliance with schedules.

Billing and Finance Manager:

1. Generate and manage invoices for clients.
2. Track payments and billing cycles.
3. Access financial records and reports.
4. Manage client billing information.
5. Communicate with clients and caregivers regarding billing issues.
6. Monitor accounts receivable and overdue payments.

Training and Compliance Officer:

1. Schedule and manage caregiver training sessions.

2. Track caregiver certifications and compliance with training requirements.
3. Access caregiver compliance records.
4. Generate compliance reports and documentation.
5. Communicate with caregivers regarding training and compliance.

HR Manager:

1. Manage caregiver recruitment and onboarding.
2. Access and update employment records.
3. Conduct performance evaluations and maintain related documentation.
4. Manage personnel files and records.
5. Communicate with caregivers regarding HR-related matters.

Quality Assurance Inspector:

1. Access caregiver documentation and care records.
2. Conduct quality assurance inspections and assessments.
3. Generate reports on caregiver performance and client satisfaction.
4. Communicate findings and recommendations to management.
5. Monitor compliance with quality standards and regulations.

IT Support:

1. Access and maintain the system's technical infrastructure.
2. Provide technical support to users.
3. Troubleshoot and resolve system issues.
4. Implement security measures and updates.
5. Manage user accounts and access privileges.

Reports Analyst:

1. Access and analyze database data.
2. Generate custom reports and analytics.
3. Share insights and findings with management.
4. Develop and maintain reporting templates.

5. List Assumptions you would like to make while designing this database

When designing a Home Care Worker Database Management System, it's important to make certain assumptions to guide the development process. These assumptions can help streamline the design and ensure that the system meets the needs of users. Here are some common assumptions:

1. **Data Privacy and Security:** Assume that the system needs to comply with strict data privacy regulations, such as HIPAA in the United States or GDPR in Europe. Implement robust security measures to protect sensitive client and caregiver information.
2. **Scalability:** Assume that the database needs to be scalable to accommodate a growing number of clients, caregivers, and records over time. Plan for efficient data storage and retrieval as the system expands.
3. **User Training:** Assume that users may require training to effectively use the system, especially caregivers who may not be tech-savvy. Design the user interface with user-friendliness in mind and consider offering training resources.
4. **Data Redundancy:** Assume that data redundancy is a concern, and design the database schema to minimize duplicate data while ensuring data consistency.
5. **Regulatory Compliance:** Assume that the home care agency must comply with local, state, and federal regulations regarding healthcare, employment, and billing. Ensure that the system can capture and report necessary compliance data.
6. **Integration with Other Systems:** Assume that the database may need to integrate with other healthcare systems, billing systems, or electronic health records (EHR) systems. Plan for interoperability and data exchange.
7. **Mobile Accessibility:** Assume that caregivers may need to access the system on mobile devices while providing care in clients' homes. Ensure the system is responsive and accessible on various devices.
8. **Backup and Disaster Recovery:** Assume the need for regular data backups and a disaster recovery plan to safeguard against data loss or system failures.
9. **Client Feedback:** Assume that clients will provide feedback on the care they receive through the system. Design mechanisms to collect and analyze this feedback to improve service quality.

10. **Care Plan Changes:** Assume that care plans for clients may change over time based on their evolving health needs. Allow for flexibility in updating care plans and associated documentation.
11. **Remote Work:** Assume that some staff members, such as schedulers or coordinators, may need remote access to the system for scheduling and coordination purposes, especially during unexpected events like pandemics.
12. **Reporting and Analytics:** Assume that users will require various reports and analytics to monitor performance, track trends, and make informed decisions. Design the database to support reporting functionality.
13. **Client and Caregiver Turnover:** Assume that there may be turnover among clients and caregivers. Ensure that the system can efficiently handle client and caregiver onboarding and offboarding processes.
14. **Multi-Language Support:** Assume that caregivers and clients may speak different languages. Provide multi-language support in the user interface and documentation.
15. **Training and Support Resources:** Assume that users will need access to training materials and support resources within the system to assist them in using the software effectively.

These assumptions should be validated and refined during the design and development process through discussions with stakeholders and careful analysis of the specific requirements of the home care agency.

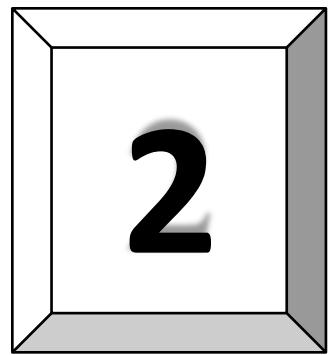
6. Business Constraints

Business constraints encompass the limitations and factors that can impact the development and operation of a Home Care Worker Database Management System. Recognizing these constraints is essential for effective project management and aligning the system with the organization's requirements. Below are some common business constraints that can influence such a system:

1. **Financial Restrictions:** The availability of financial resources may restrict the system's scope and features. The project must adhere to the allocated budget, potentially affecting technology choices, staffing, and project timelines.
2. **Time and Deadlines:** Tight schedules for system development and deployment could be a factor. Regulatory obligations or business urgencies might necessitate a swift implementation, potentially affecting system complexity and testing time.
3. **Regulatory Adherence:** Compliance with healthcare regulations and privacy laws is a substantial constraint. Non-compliance could result in legal consequences and penalties, requiring the system to be meticulously designed and maintained to meet these standards.
4. **Staffing and Expertise:** The availability of proficient IT personnel and domain specialists, such as healthcare and home care experts, can be limited. A scarcity of qualified staff can impact system development and maintenance.
5. **Integration Needs:** If integration with existing healthcare systems, electronic health records (EHR), or billing systems is necessary, constraints related to compatibility and data exchange protocols should be considered.
6. **Data Privacy and Security:** Strict adherence to data privacy and security standards is imperative. Constraints linked to encryption, access controls, and data breach prevention can significantly influence system design and operation.
7. **User Adoption:** Ensuring that clients and caregivers can effectively use the system may be a constraint. Factors like user-friendliness, training, and support can influence user acceptance.
8. **Scalability:** The system should have the ability to expand to accommodate a growing clientele, caregivers, and records. Constraints pertaining to server capacity and database scalability must be accounted for.
9. **Client Diversity:** The varying demographics of clients, encompassing age, health conditions, and technological proficiency, can be a constraint. The system should be versatile enough to accommodate diverse client needs and preferences.

10. **Matching Caregivers with Clients:** Efficiently matching caregivers with clients based on skills, availability, and preferences can be a challenge, especially as the client and caregiver pool expands.
11. **Mobile Access:** Enabling mobile access for caregivers and staff who require on-the-go system access can be a constraint, as it necessitates responsive design and mobile app development.
12. **Change Management:** Implementing a new system may necessitate changes in workflows and processes. Staff resistance to change could be a constraint that requires effective management.
13. **Client Feedback and Satisfaction:** Gathering client feedback and ensuring their satisfaction can be a constraint if not adequately addressed. The system should facilitate feedback collection and analysis.
14. **Geographic Reach:** If the home care agency operates across a wide geographic area, constraints related to service coverage, transportation, and caregiver availability in specific locations should be taken into account.
15. **Competitive Environment:** The presence of competitors in the home care industry can be a constraint. The system might need to offer unique features or enhance efficiency to remain competitive.

These business constraints should be carefully considered and incorporated into the planning, design, and implementation phases of the Home Care Worker Database Management System to ensure it effectively aligns with the organization's objectives.



Section : ER Diagram

1. Noun (& Verb) Analysis.

Nouns	Verbs
Home	has become
Care	face
Database	must address
Management	ensuring
System	streamlining
Era	includes
Population	generates
Demand	tracks
Healthcare	provides
Role	simplifying
Services	streamlines
Support	track
Individuals	identify
Need	offers
Agencies	generate
Caregiver	integrate
Information	safeguarding
Web	instills
Schedules	provide
Compliance	facilitates
Challenges	enhancing
Quality	enables
Repository	reducing
Caregivers	promotes
Demographic	collecting
Data	translate
Histories	empower
Plans	optimize
Availability	improve
Preferences	allow
Process	creating

Allocation	manages
Invoices	manage
Payments	assigns
Insights	enabling
Processes	assigned
Agencies	sent
Metrics	record
Feedback	involved
Mechanisms	supports
Indicators	facilitating
Areas	catering
Reporting	taken
Analytics	assists
Performance	helping
Satisfaction	provide
Features	simplifies
Interface	reports
Efficiency	exported
Decisions	empowers
Stack	making
Measures	taking
Encryption	evolved
Control	extending
Integrity	recognized
Scalability	stands
Growth	driving
Number	offered
Services	providing
Array	receiving
Minimum	taking
Areas	interact
Meal	interact
Planning	view
Medication	view
	...

Toileting	offering
Mobility	ensuring
Support	ensuring
Management	impacting
Monitoring	recognized
Nutrition	stands
Reminders	driving
Housekeeping	designed
Cleaning	address
Laundry	provide
Organization	improve
Transportation	ensuring
Appointments	protect
Outings	optimize
Relief	facilitate
Training	ensure
Palliative	simplify
Comfort	prioritize
Pain	overcome
Emotional	improve
Illnesses	deliver
Families	ensure
Variety	elevate
Breadth	meet
Needs	uphold
Challenges	ensure
Delivery	provide
Role	affect
Healthcare	provide
Mode	evolve
Challenges	stands
Agencies	drive
Coordination	empower
Services	address
Client	protect
Range	optimize
Professionals	offer
Services	facilitate

Information	ensure
Obligation	simplify
Measures	prioritize
Access	
Clients	
Skills	
Requirements	
Mismatch	
Turnover	
Understanding	
Policies	
Expenses	
Rates	
Guidance	
Support	
Communication	
Messages	
Caregivers	
Misunderstandings	
Errors	
Delivery	
Response	
Situations	
Agencies	
Compliance	
Industry	
Regulations	
Standards	
Agencies	
Records	
Records	
Consequences	
Repercussions	
Solutions	
Coordination	
Agencies	
Schedules	

Conflicts	
Appointments	
Experience	
Protection	
Measures	
Encryption	
Control	
Requirements	
Information	
Commitment	
Security	
Challenges	
Delivery	
Solutions	
Coordination	
Privacy	
Guidance	
Communication	
Features	
Messaging	
Reminders	
Plan	
Protocols	
Emergencies	
Response	
Situation	
Agencies	
Compliance	
Healthcare	
Regulations	
Documentation	
Compliance	
Risks	
Data	
Measures	
Encryption	
Access	
Backups	

Information	
Asset	
Data	
Challenges	
Solutions	
Care	
Coordination	
System	
Challenges	
Solutions	
Agencies	
Clients	
Experience	
Caregivers	
Services	
Clients	
Needs	
Agencies	
Client	
Information	
Processes	
Reports	
Reports	
Data	
Solutions	
Challenges	
Compliance	
System	
Protocols	
Caregivers	
Resources	
Situations	
Emergencies	
Caregivers	
Resources	
Guidance	
Options	
Stress	
Transparency	

Communication System	
Communication Features	
Messaging	
Reminders	
Plan	
Protocols	
Response	
Situation	
Agencies	
Compliance	
Healthcare	
Regulations	
Documentation	
Compliance	
Risks	
Data	
Measures	
Encryption	
Access	
Backups	
Information	
Asset	
Data	

Challenges	
Solutions	
Care	
Coordination	
System	
Challenges	
Solutions	
Agencies	
Clients	
Experience	
Caregivers	
Services	
Clients	
Needs	
Agencies	
Client	
Information	
Processes	
Reports	
Reports	
Data	
Solutions	
Challenges	
Compliance	
System	
Protocols	
Caregivers	

Resources	
Situations	
Emergencies	
Caregivers	
Resources	
Guidance	
Options	
Stress	
Transparency	
Communication System	
Communication	
Features	
Messaging	
Reminders	
Plan	
Protocols	
Response	
Situation	
Agencies	
Compliance	
Healthcare	
Regulations	
Documentation	

2. Create Table.1. & Table.2. as per the below-given format for accepted nouns list

Table 1:

Noun	Inclusion Reason	Response	Necessary for reacting to situations.
Organization	Represents entities involved in the system.	Situations	Represents various conditions.
Mode	Describes different operation modes.	Documentation	Essential for recording and record-keeping.
Coordination	Essential for managing activities.	Risks	Potential issues and problems.
Obligation	Pertains to legal and ethical responsibilities.	Data	Essential for system operation and management.
Obligations	Pertains to legal and ethical responsibilities.	Asset	Crucial for resource management.
Role	Describes different positions or responsibilities.	Challenges	Describes potential difficulties in the domain.
Services	Essential offerings within the domain.	Solutions	Represent potential resolutions.
Information	Fundamental data needed for the system.	Care	Represents a central concept in the domain.
Measures	Essential for assessing performance.	Coordination	Essential for managing activities.
Access	Key for using and protecting data.	System	The core element of the system.
Skills	Represents caregiver capabilities.	Protocols	Specific procedures and guidelines.
Requirements	Specific criteria that must be met.	Resources	Fundamental elements for operation.
Turnover	Describes staff changes.	Stress	Represents potential strain.
Understanding	Pertains to knowledge and comprehension.	Transparency	Needed for openness and clarity.
Policies	Rules and guidelines governing the system.	Features	Attributes of the system.
Expenses	Represents costs and financial aspects.	Messaging	Communication within the system.
Rates	Pertains to pricing and charges.	Reminders	Essential for task management.
Guidance	Necessary for providing direction.	Plan	A fundamental part of care strategies.
Support	Essential for assistance and help.	Options	Provides choices and alternatives.
Communication	Crucial for information exchange.	Situation	Represents various conditions.
Messages	Information transmitted between parties.	Industry	Pertains to the broader context of care.
Misunderstandings	Represents potential confusion.	Regulations	Legal and industry requirements.
Errors	Describes mistakes and issues.	Standards	Expected quality levels.
Delivery	Part of providing services.	Records	Necessary for documentation and history.
		Consequences	Outcomes and results.

Repercussions	Outcomes and results.
Solutions	Represent potential resolutions.
Coordination	Essential for managing activities.
Schedules	Necessary for planning and coordination.
Conflicts	Describes potential disagreements.
Appointments	Specific scheduling needs.
Experience	Past involvement in the domain.
Protection	Necessary for safeguarding data.
Measures	Essential for assessing performance.
Encryption	Key for data security.
Control	Fundamental for system operation.
Requirements	Specific criteria that must be met.
Information	Fundamental data needed for the system.
Commitment	Represents dedication and obligation.
Security	Crucial for protecting data and systems.
Challenges	Describes potential difficulties in the domain.

Delivery	Part of providing services.
Solutions	Represent potential resolutions.
Coordination	Essential for managing activities.
Privacy	Pertains to data and information protection.
Guidance	Necessary for providing direction.
Communication	Crucial for information exchange.
Features	Attributes of the system.
Messaging	Communication within the system.
Reminders	Essential for task management.
Plan	A fundamental part of care strategies.
Protocols	Specific procedures and guidelines.
Response	Necessary for reacting to situations.
Situation	Represents various conditions.
Agencies	Relevant organizations in the context.
Compliance	Key for adhering to regulations and standards.
Healthcare	Pertains to the broader context of care.

Table 2 :

Attribute Name	Likely Entity Set to be Assigned	Caregiver Preferences	Preferences, Caregiver
		Medical Conditions	Medical, Client
Caregiver ID	Caregiver	Medication Schedule	Schedule, Medication
Client ID	Client	Payment Options	Options, Payment
Employee ID	Employee	Performance Reviews	Reviews, Caregiver
Service Date	Service, Schedule	Service Ratings	Ratings, Service
Service Time	Service, Schedule	Incident Reports	Reports, Incident
Service Location	Service, Location	Geographic Service Area	Area, Service
Payment Records	Payment	Languages Spoken	Languages, Caregiver
Billing Period	Billing, Schedule	Work Shifts	Shifts, Schedule
Invoice Details	Billing, Invoice	Availability Calendar	Calendar, Schedule
Compliance Records	Compliance	Equipment Inventory	Inventory, Equipment
Care Plan Updates	Care Plan	Caregiver Certifications	Certifications, Caregiver
Caregiver Training Records	Training, Caregiver	Marketing Data	Marketing
Caregiver Availability	Schedule, Caregiver	Transportation Options	Transportation
Client Appointments	Schedule, Client	Meal Planning	Meal Planning
Emergency Contact Information	Contact, Emergency	Service Logs	Service Logs
Client Preferences	Preferences, Client	Employee Training Records	Training, Employee

3.

Create Table.3. as per the below-given format for rejected nouns list

Table 3 :

Excluded Word	Exclusion Reason
Asset	Duplicates
Client	Duplicates
Clients	Duplicates
Communication	Duplicates
Coordination	Duplicates
Data	Duplicates
Documentation	Duplicates
Emergencies	Duplicates
Encryption	Duplicates
Measures	Duplicates
Protocols	Duplicates
Regulations	Duplicates
Reports	Duplicates
Response	Duplicates
Situations	Duplicates
Solutions	Duplicates
System	Duplicates
Features	Duplicates
Messaging	Duplicates

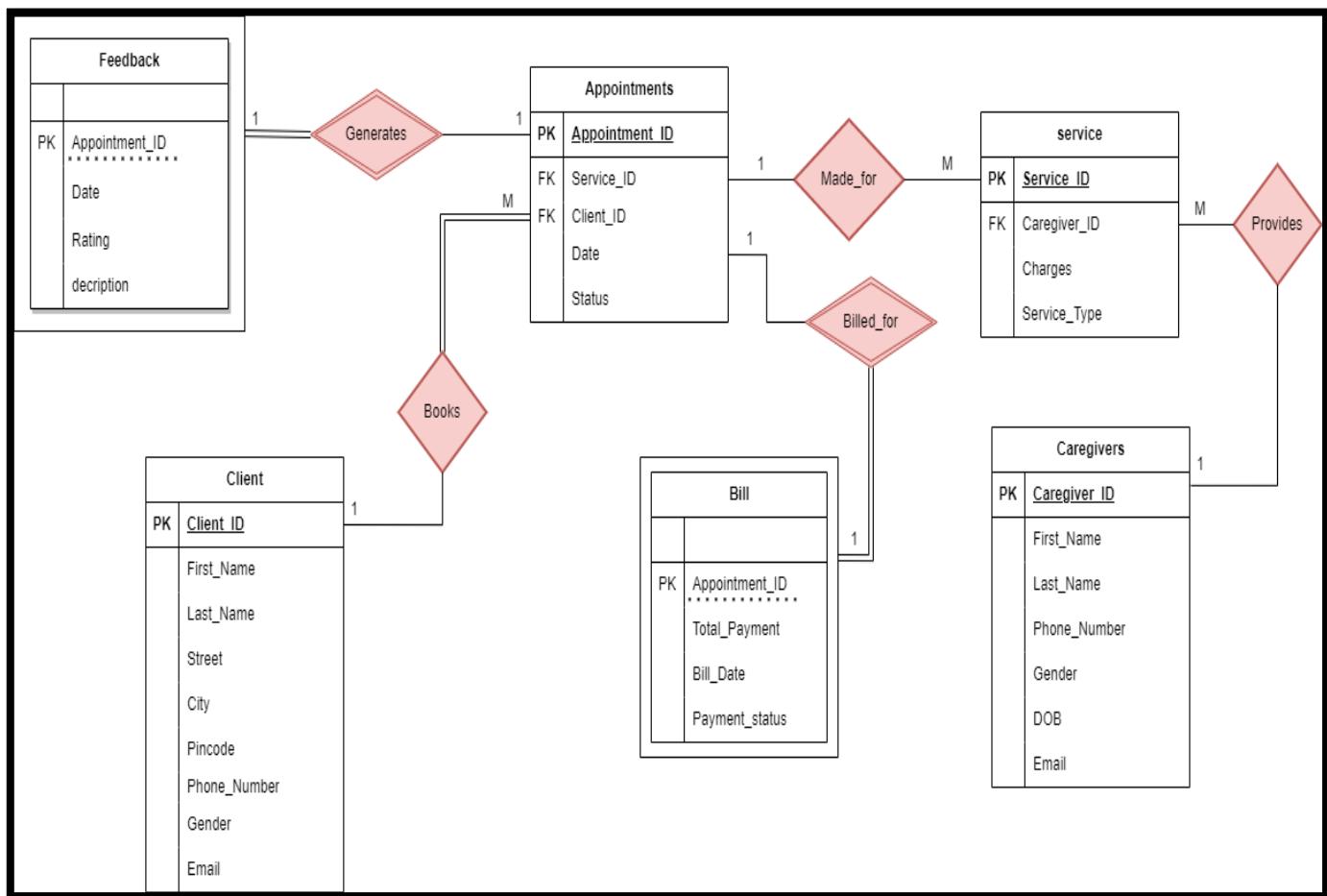
Reminders	Duplicates
Protocols	Duplicates
Response	Duplicates
Situation	Duplicates
Area	Irrelevant
Era	Irrelevant
Population	Irrelevant
Demand	Irrelevant
Era	Irrelevant
Population	Irrelevant
Demand	Irrelevant
Individuals	Irrelevant
Need	Irrelevant
Web	Irrelevant
Schedules	Irrelevant
Challenges	Irrelevant
Repository	Irrelevant
Tools	Irrelevant
Conflicts	Irrelevant
Resource	Irrelevant
Invoices	Irrelevant
Payments	Irrelevant

Insights	Irrelevant
Processes	Irrelevant
Areas	Irrelevant
Integration	Irrelevant
Security	Irrelevant
Access	Irrelevant
Growth	Irrelevant
Number	Irrelevant
Services	Irrelevant
Array	Irrelevant
Minimum	Irrelevant
Wound	Vague
Toileting	Vague
Support	Vague
Management	General
Monitoring	General
Client	Attributes
Histories	Attributes
Plans	Attributes
Availability	Attributes
Preferences	Attributes
Process	Associations

Final Entity Set Table :

Client	Client_ID (Primary Key) <ul style="list-style-type: none"> - First_Name - Last_Name - Street - City - Pincode - Phone_NUmber - Gender - Email
Caregiver	CaregiverID (Primary Key) <ul style="list-style-type: none"> - First_Name - Last_Name - Phone_Number - DOB - Gender - Email
Service	Service_ID (Primary Key) <ul style="list-style-type: none"> - Service Type (e.g., Medical Care, Personal Care , Meal Preparation) - Caregiver_ID (FK) - Charges
Appointments	Appointment_ID(Primary Key) <ul style="list-style-type: none"> - Client_ID (Foreign Key) - Service_ID (Foreign Key) - Date - Status
Bill	<ul style="list-style-type: none"> - Appointment_ID (Partial Key) - Total_Payment - Bill_Date - Payment_status
Feedback	<ul style="list-style-type: none"> - Appointment_Id(Partial key) - Date - Rating - Description

2. Develop the ER Diagram (ERD).



Client Table

Client_ID	PK
First_Name	Varchar(25) NOT NULL
Last_Name	Varchar(25) NOT NULL
Street	Varchar(15) NOT NULL
City	Varchar(15) NOT NULL
Pincode	Int(10) NOT NULL
Phone_Number	Int(15) NOT NULL
Email	Varchar(255)
Gender	Varchar(15) NOT NULL check(Gender = Male or Gender = Female)

Caregivers Table

Caregiver_ID	PK
First_Name	Varchar(25) NOT NULL
Last_Name	Varchar(25) NOT NULL
DOB	Varchar(15) NOT NULL
Phone_Number	Int(15) NOT NULL
Gender	Varchar(15) NOT NULL check(Gender = Male or Gender = Female)
Email	Varchar(255)

Service Table

Service_ID	PK
Caregiver_ID	FK
Charges	Int(255) NOT NULL
Service_Type	Varchar(255) NOT NULL

Appointments Table

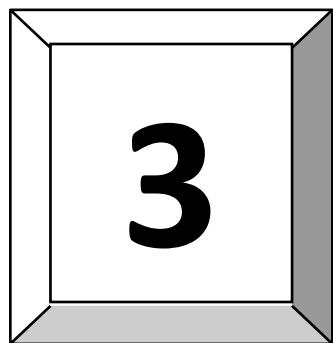
Appointment_ID	PK
Service_ID	FK
Client_ID	FK
Date	Varchar(10) NOT NULL
Status	Varchar(15) NOT NULL check(Status = Complete or Status = Pending or Status = Rejected)

Feedback Table

Appointment_ID	FK
Date	Varchar(10) NOT NULL
Rating	int(10) NOT NULL
description	Varchar(255) NOT NULL

Bill Table

Appointment_ID	FK
Total_Payment	int(255) NOT NULL
Bill_Date	Varchar(15) NOT NULL
Payment_status	Varchar(15) NOT NULL check(Status = Done or Status = Pending)



Section: . Database Schema

1. Mapping E-R Model to Relational Model

- **Client**(Client_id,first_name,last_name,street,city,pincode,phone_number,gender,email)
- **Caregivers**(caregiver_id,first_name,last_name,skill,DOB,phone_id,gender)
- **Service**(service_id,cargiver_id,charges,service_type)
- **Appointments**(appointment_id,service_id,client_id,date,status)
- **Feedback**(appointment_id,Date,rating,description)
- **Bill**(appointment_id,total_payment,bill_date,payment_status)

3. Normalization & Schema Refinement

a). List of redundancies existing for every schema which is part of the database.

Clients (Table):

Contact Information: If a client can have multiple phone numbers these should be stored in a separate "Contact Information" table with a foreign key relationship to the "Clients" table. This avoids repeating groups of contact information.

Caregivers (Table):

Caregivers Information: If a caregiver can have multiple phone numbers these should be stored in a separate "Contact Information" table with a foreign key relationship to the "caregiver" table. This avoids repeating groups of contact information

C). Document the logic of how you arrived at the 3NF/BCNF design step by step, starting from the original design.

1) Clients (Table):

Client(Client_id,first_name,last_name,street,city,pincode,phone_number,gender,email)

Step 1: 1NF (First Normal Form)

Ensure that there are no repeating groups or arrays, and each attribute has a single value. To convert the Client table to the first normal form (1NF) while treating the "Phone_Number" attribute as a multi-valued attribute, we need to create a separate table for phone numbers and link it to the Client table using a foreign key. Here's the proper format for the conversion :

Client(Client_id,first_name,last_name,street,city,pincode,phone_number,gender,email)

Client_Contact(Client_id, phone_number)

Step 2: 2NF (Second Normal Form)

Ensure that there are no partial dependencies on the primary key. The Client table is already in the second normal form (2NF) because all non-key attributes are fully functionally dependent on the entire primary key (**Client_ID**).

Step 3: 3NF (Third Normal Form)

Ensure that there are no transitive dependencies.

A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.

BCNF (Boyce-Codd Normal Form) :

Check for non-trivial functional dependencies where a non-prime attribute determines another non-prime attribute. In this case, if there are no such dependencies, the table is already in BCNF.

2) Caregiver Table

caregivers(caregiver_id,first_name,last_name,skill,DOB,phone_number,gender)

Step 1: 1NF (First Normal Form)

Ensure that there are no repeating groups or arrays, and each attribute has a single value. To convert the **Caregiver** table to the first normal form (1NF) while treating the "Phone_Number" attribute as a multi-valued attribute, we need to create a separate table for phone numbers and link it to the **Caregiver** table using a foreign key. Here's the proper format for the conversion:

Caregiver_Contact(Caregiver_id, phone_number)

Step 2: 2NF (Second Normal Form)

Ensure that there are no partial dependencies on the primary key.

The **Caregiver** table is already in the second normal form (2NF) because all non-key attributes are fully functionally dependent on the entire primary key (**Caregiver**).

Step 3: 3NF (Third Normal Form)

Ensure that there are no transitive dependencies.

A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.

This table is already in 3nf and bcnf

3) Service Table

service(service_id,cargiver_id,charges,service_type)

Step 1: 1NF (First Normal Form)

Ensure that there are no repeating groups or arrays, and each attribute has a single value

service(service_id,service_type_id,cargiver_id,charges)

This table is already in 1nf

Step 2:2NF (Second Normal Form):

2NF requires that there are no partial dependencies. In this case, we already have a candidate key (Service_ID) and no partial dependencies, so the table is in 2NF.

This table is already in 2nf

Step 3: 3NF (Third Normal Form)

Ensure that there are no transitive dependencies.

A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.

This table is already in 3nf

Step 4:BCNF (Boyce-Codd Normal Form)

To achieve BCNF, every non-trivial functional dependency should have the left-hand side as a superkey. In the case of the Service table, it is already in BCNF as there are no non-trivial partial or transitive dependencies.

service(service_id,service_type_id,cargiver_id,charges)

4)Appointments Table

appointments(appointment_id,service_id,client_id,date,status)

Step 1: 1NF (First Normal Form)

Ensure that there are no repeating groups or arrays, and each attribute has a single value.
The table is already in 1NF.

Step 2:2NF (Second Normal Form):

2NF requires that there are no partial dependencies

appointments(appointment_id,service_id,client_id,date)

The table is already in 2NF.

Step 3: 3NF (Third Normal Form)

Ensure that there are no transitive dependencies.

A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.

This table is already in 3nf

5) Bill Table

Step 1: 1NF (First Normal Form)

Ensure that there are no repeating groups or arrays, and each attribute has a single value

The table is already in 1NF.

Step 2: 2NF (Second Normal Form):

this table is already in 2nf

Step 3: 3NF (Third Normal Form)

Ensure that there are no transitive dependencies.

there are no transitive dependencies, and all attributes are functionally dependent on the primary key (**Bill_ID** and **Appointment_ID**). Therefore, the table is in the third normal form (3NF).

6) Feedback Table

Step 1: 1NF (First Normal Form)

Ensure that there are no repeating groups or arrays, and each attribute has a single value

The table is already in 1NF.

Step 2: 2NF (Second Normal Form):

This table is already in 2nf

Step 3: 3NF (Third Normal Form)

Ensure that there are no transitive dependencies.

There are no transitive dependencies, and all attributes are directly dependent on the primary key, which consists of both **Appointment_ID** and **Date**. Therefore, the table is in the third normal form (3NF).

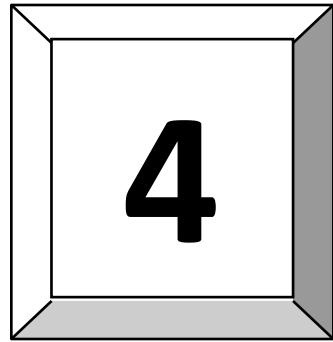
Step 4: BCNF (Boyce-Codd Normal Form)

To achieve BCNF, every non-trivial functional dependency should have the left-hand side as a superkey. In the case of the Service table, it is already in BCNF as there are no non-trivial partial or transitive dependencies.

The **Feedback** table is in Boyce-Codd Normal Form (BCNF), as it satisfies the requirement that for every non-trivial functional dependency, the determinant (X) is a superkey.

4. Final Relation Schema

- Client(Client_id,first_name,last_name,street,city,pincod,gender,email)
- Client_Contact(Client_id, phone_number)
- caregivers(caregiver_id,first_name,last_name,skill,DOB ,gender)
- Caregiver_Contact(Caregiver_id, phone_number)
- service(service_id,cargiver_id,charges,service_type)
- appointments(appointment_id,service_id,client_id,date,status)
- feedback(appointment_id,Date,rating,description)
- bill(appointment_id,total_payment,bill_date,payment_status)



Section: DDL and SQL

1. Create DDL Scripts

i. Express the following constraints (if applicable) in DDL:

1. Domain constraints
2. Key constraints
3. Referential Integrity Constraints
4. Any other constraints

Client Table	
Client_ID	PK
First_Name	Varchar(25) NOT NULL
Last_Name	Varchar(25) NOT NULL
Street	Varchar(15) NOT NULL
City	Varchar(15) NOT NULL
Pincode	Int(10) NOT NULL
Email	Varchar(255)
Gender	Varchar(15) NOT NULL check(Gender = Male or Gender = Female)

Client_Contact	
Client_id	Fk
phone_number	int(255) NOT NULL

Caregivers Table	
Caregiver_ID	PK
First_Name	Varchar(25) NOT NULL
Last_Name	Varchar(25) NOT NULL
DOB	Varchar(15) NOT NULL
Gender	Varchar(15) NOT NULL check(Gender = Male or Gender = Female)
Email	Varchar(255)

Caregiver_Contact	
Caregiver_id	PK
phone_number	int(255) NOT NULL

Service Table	
Service_ID	PK
Caregiver_ID	FK
Charges	Int(255) NOT NULL
Service_Type	Varchar(255) NOT NULL

Appointments Table	
Appointment_ID	PK
Service_ID	FK
Client_ID	FK
Date	Varchar(10) NOT NULL
Status	Varchar(15) NOT NULL check(Status = Complete or Status = Pending or Status = Rejected)

Feedback Table	
Appointment_ID	PK
Date	Varchar(10) NOT NULL
Rating	int(10) NOT NULL
description	Varchar(255) NOT NULL

Bill Table	
Appointment_ID	PK
Total_Payment	int(255) NOT NULL
Bill_Date	Varchar(15) NOT NULL
Payment_status	Varchar(15) NOT NULL check(Status = Done or Status = Pending)

1. Write DDL Scripts.

- Client Table

```
CREATE TABLE "Homecare".Client
(
    "Client_Id" character varying NOT NULL,
    "First_Name" character varying NOT NULL,
    "Last_Name" character varying NOT NULL,
    "Street" character varying NOT NULL,
    "City" character varying NOT NULL,
    "Pincode" integer NOT NULL,
    "Email" character varying,
    "Gender" character varying NOT NULL,
    PRIMARY KEY ("Client_Id")
);
```

- Client Contact Table

```
CREATE TABLE "Homecare".Client_Contact
(
    "Client_Id" character varying NOT NULL,
    "Phone_Number" integer NOT NULL,
    CONSTRAINT "Client_Id" FOREIGN KEY ("Client_Id")
        REFERENCES "Homecare".client ("Client_Id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

- Caregiver Table

```
CREATE TABLE "Homecare".Caregivers
(
    "Caregiver_Id" character varying,
    "First_Name" character varying NOT NULL,
    "Last_Name" character varying NOT NULL,

    "Dob" date NOT NULL,
    "Gender" character varying NOT NULL,
    "Email" character varying,
    PRIMARY KEY ("Caregiver_Id"),
    CONSTRAINT "Gender" CHECK ("Gender" in('male','female')) NOT VALID
);
```

- Caregiver Contact Table

```
CREATE TABLE "Homecare".caregiver_Contact
(
    "Caregiver_Id" character varying NOT NULL,
    "Phone_Number" integer NOT NULL,
    CONSTRAINT "Caregiver_Id" FOREIGN KEY ("Caregiver_Id")
        REFERENCES "Homecare".caregivers ("Caregiver_Id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

- Service Table

```
CREATE TABLE "Homecare".Service
(
    "Service_Id" character varying,
    "Caregiver_Id" character varying NOT NULL,
    "Charges" integer NOT NULL,
    "Service_Type" character varying NOT NULL,
    PRIMARY KEY ("Service_Id"),
    CONSTRAINT "Caregiver_Id" FOREIGN KEY ("Caregiver_Id")
        REFERENCES "Homecare".caregivers ("Caregiver_Id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

- Appointments

```

CREATE TABLE "Homecare".Appointments
(
    "Appointment_Id" character varying,
    "Service_Id" character varying NOT NULL,
    "Client_Id" character varying NOT NULL,
    "Date" date NOT NULL,
    "Status" character varying NOT NULL,
    PRIMARY KEY ("Appointment_Id"),
    CONSTRAINT "Service_Id" FOREIGN KEY ("Service_Id")
        REFERENCES "Homecare".Service ("Service_Id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Client_ID" FOREIGN KEY ("Client_Id")
        REFERENCES "Homecare".Client ("Client_Id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Status" CHECK ("Status" in('complete','pending','rejected')) NOT
    VALID
);

```

- Bill Table

```

CREATE TABLE "Homecare".bill
(
    "Apppointment_id" character varying,
    "Total_payment" integer NOT NULL,
    "Bill_Date" date NOT NULL,
    "Payment_status" character varying NOT NULL,
    CONSTRAINT "Apppointment_id" FOREIGN KEY ("Apppointment_id")
        REFERENCES "Homecare".Appointments ("Appointment_Id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Payment_status" CHECK ("Payment_status" in('done','pending'))
    NOT VALID
);

```

- Feedback Table

```
CREATE TABLE "Homecare".Feedback
(
    "Appointment_id" character varying,
    "Date" date NOT NULL,
    "Rating" integer NOT NULL,
    description character varying NOT NULL,
    CONSTRAINT "Appointment_ID" FOREIGN KEY ("Appointment_id")
        REFERENCES "Homecare".Appointments ("Appointment_Id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

```
INSERT INTO "Homecare".client ("Client_Id", "First_Name", "Last_Name", "Street", "City",
"Pincode", "Email", "Gender") VALUES
('c1', 'Aarav', 'Kumar', 'Surya Nagar', 'Mumbai', 400001, 'aarav.kumar@example.com', 'male'),
('c2', 'Aarya', 'Sharma', 'Lotus Colony', 'Delhi', 110001, 'aarya.sharma@example.com', 'female'),
('c3', 'Aditi', 'Joshi', 'Rose Street', 'Bangalore', 560001, 'aditi.joshi@example.com', 'female'),
('c4', 'Amit', 'Patel', 'River View', 'Ahmedabad', 380001, 'amit.patel@example.com', 'male'),
('c5', 'Anaya', 'Singh', 'Golden Avenue', 'Chennai', 600001, 'anaya.singh@example.com',
'female'),
('c6', 'Arjun', 'Gupta', 'Heritage Road', 'Kolkata', 700001, 'arjun.gupta@example.com', 'male'),
('c7', 'Avni', 'Shah', 'Diamond Lane', 'Hyderabad', 500001, 'avni.shah@example.com', 'female'),
('c8', 'Dev', 'Verma', 'Silver Street', 'Pune', 411001, 'dev.verma@example.com', 'male'),
('c9', 'Esha', 'Rao', 'Jade Lane', 'Surat', 395001, 'esha.rao@example.com', 'female'),
('c10', 'Ishaan', 'Malhotra', 'Emerald Road', 'Jaipur', 302001, 'ishaan.malhotra@example.com',
'male'),
('c11', 'Kabir', 'Deshpande', 'Topaz Lane', 'Lucknow', 226001, 'kabir.deshpande@example.com',
'male'),
('c12', 'Kavya', 'Reddy', 'Amber Street', 'Nagpur', 440001, 'kavya.reddy@example.com',
'female'),
('c13', 'Leela', 'Gandhi', 'Sapphire Lane', 'Indore', 452001, 'leela.gandhi@example.com',
'female'),
('c14', 'Mira', 'Chatterjee', 'Platinum Avenue', 'Vadodara', 390001,
'mira.chatterjee@example.com', 'female'),
('c15', 'Neha', 'Bose', 'Pearl Street', 'Kanpur', 208001, 'neha.bose@example.com', 'female'),
('c16', 'Om', 'Dixit', 'Ruby Lane', 'Bhopal', 462001, 'om.dixit@example.com', 'male'),
('c17', 'Pari', 'Sharma', 'Opal Street', 'Patna', 800001, 'pari.sharma@example.com', 'female'),
('c18', 'Pranav', 'Iyer', 'Coral Lane', 'Ludhiana', 141001, 'pranav.iyer@example.com', 'male'),
```

('c19', 'Riya', 'Chopra', 'Amethyst Road', 'Agra', 282001, 'riya.chopra@example.com', 'female'),
('c20', 'Rohan', 'Srinivasan', 'Jasper Avenue', 'Kochi', 682001, 'rohan.srinivasan@example.com',
'male'),
('c21', 'Saanvi', 'Khan', 'Emerald Lane', 'Varanasi', 221001, 'saanvi.khan@example.com',
'female'),
('c22', 'Shaurya', 'Nair', 'Silver Lane', 'Madurai', 625001, 'shaurya.nair@example.com', 'male'),
('c23', 'Shreya', 'Venkatesh', 'Gold Street', 'Coimbatore', 641001,
'shreya.venkatesh@example.com', 'female'),
('c24', 'Tanvi', 'Prasad', 'Ruby Street', 'Allahabad', 211001, 'tanvi.prasad@example.com',
'female'),
('c25', 'Vivaan', 'Saxena', 'Topaz Avenue', 'Amritsar', 143001, 'vivaan.saxena@example.com',
'male'),
('c26', 'Yash', 'Kumar', 'Sapphire Street', 'Srinagar', 190001, 'yash.kumar@example.com',
'male'),
('c27', 'Zara', 'Jain', 'Platinum Lane', 'Jodhpur', 342001, 'zara.jain@example.com', 'female'),
('c28', 'Zoya', 'Venkataraman', 'Diamond Street', 'Ghaziabad', 201001,
'zoya.venkataraman@example.com', 'female'),
('c29', 'Aadi', 'Dubey', 'Coral Avenue', 'Firozabad', 283203, 'aadi.dubey@example.com', 'male'),
('c30', 'Ahaan', 'Bhattacharya', 'Amethyst Lane', 'Bhubaneswar', 751001,
'ahaan.bhattacharya@example.com', 'male'),
('c31', 'Amaira', 'Ghosh', 'Jade Avenue', 'Bikaner', 334001, 'amaira.ghosh@example.com',
'female'),
('c32', 'Anika', 'Rao', 'Emerald Street', 'Gorakhpur', 273001, 'anika.rao@example.com', 'female'),
('c33', 'Ayaan', 'Chowdhury', 'Silver Avenue', 'Guntur', 522001,
'ayaan.chowdhury@example.com', 'male'),
('c34', 'Darsh', 'Sethi', 'Gold Lane', 'Haridwar', 249401, 'darsh.sethi@example.com', 'male'),
('c35', 'Ira', 'Gupta', 'Ruby Avenue', 'Jamnagar', 361001, 'ira.gupta@example.com', 'female'),
('c36', 'Kiaan', 'Prasad', 'Topaz Street', 'Kurnool', 518001, 'kiaan.prasad@example.com', 'male'),
('c37', 'Kyra', 'Singh', 'Sapphire Avenue', 'Mathura', 281001, 'kyra.singh@example.com',
'female'),
('c38', 'Manan', 'Chopra', 'Platinum Street', 'Muzaffarnagar', 251001,
'manan.chopra@example.com', 'male'),
('c39', 'Myra', 'Shah', 'Diamond Avenue', 'Nanded', 431601, 'myra.shah@example.com',
'female'),
('c40', 'Navya', 'Malhotra', 'Coral Street', 'Nashik', 422001, 'navya.malhotra@example.com',
'female'),
('c41', 'Nivaan', 'Deshpande', 'Amethyst Avenue', 'Ratlam', 457001,
'nivaan.deshpande@example.com', 'male'),
('c42', 'Reyansh', 'Sinha', 'Jade Street', 'Rourkela', 769001, 'reyansh.sinha@example.com',
'male'),
('c43', 'Rishaan', 'Chaudhary', 'Emerald Avenue', 'Sagar', 470002,
'rishaan.chaudhary@example.com', 'male'),
('c44', 'Saisha', 'Nair', 'Silver Street', 'Shimla', 171001, 'saisha.nair@example.com', 'female'),
('c45', 'Shlok', 'Venkataraman', 'Gold Avenue', 'Thanjavur', 613001,

```

'shlok.venkataraman@example.com', 'male'),
('c46', 'Vihaan', 'Ghosh', 'Ruby Street', 'Tirunelveli', 627001, 'vihaan.ghosh@example.com',
'male'),
('c47', 'Advika', 'Rao', 'Topaz Avenue', 'Tirupati', 517501, 'advika.rao@example.com', 'female'),
('c48', 'Ahaana', 'Chowdhury', 'Sapphire Street', 'Tiruppur', 641601,
'ahaana.chowdhury@example.com', 'female'),
('c49', 'Amita', 'Sethi', 'Platinum Avenue', 'Udaipur', 313001, 'amita.sethi@example.com',
'female'),
('c50', 'Arohi', 'Gupta', 'Diamond Lane', 'Ujjain', 456010, 'arohi.gupta@example.com', 'female');

```

```

INSERT INTO "Homecare".Client ("Client_Id", "First_Name", "Last_Name", "Street", "City",
"Pincode", "Email", "Gender") VALUES
('c51', 'Aarna', 'Kulkarni', 'Coral Street', 'Aurangabad', 431001, 'aarna.kulkarni@example.com',
'female'),
('c52', 'Aaryan', 'Gupta', 'Amethyst Avenue', 'Bareilly', 243001, 'aaryan.gupta@example.com',
'male'),
('c53', 'Ahaana', 'Sharma', 'Jade Street', 'Belagavi', 590001, 'ahaana.sharma@example.com',
'female'),
('c54', 'Amayra', 'Chatterjee', 'Emerald Avenue', 'Bhavnagar', 364001,
'amayra.chatterjee@example.com', 'female'),
('c55', 'Anay', 'Srinivasan', 'Silver Street', 'Bhilai', 490006, 'anay.srinivasan@example.com',
'male'),
('c56', 'Aradhya', 'Venkatesh', 'Gold Avenue', 'Bokaro', 827001,
'aradhya.venkatesh@example.com', 'female'),
('c57', 'Dhruv', 'Prasad', 'Ruby Street', 'Durgapur', 713201, 'dhruv.prasad@example.com',
'male'),
('c58', 'Hridaan', 'Malhotra', 'Topaz Avenue', 'Gaya', 823001, 'hridaan.malhotra@example.com',
'male'),
('c59', 'Inaaya', 'Deshpande', 'Sapphire Street', 'Gopalpur', 760002,
'inaaya.deshpande@example.com', 'female'),
('c60', 'Ishan', 'Rao', 'Platinum Avenue', 'Gulbarga', 585101, 'ishan.rao@example.com', 'male'),
('c61', 'Ivaan', 'Chowdhury', 'Diamond Lane', 'Guna', 473001, 'ivaan.chowdhury@example.com',
'male'),
('c62', 'Kabir', 'Sethi', 'Coral Street', 'Haldwani', 263139, 'kabir.sethi@example.com', 'male'),
('c63', 'Kaira', 'Shah', 'Amethyst Avenue', 'Haridwar', 249401, 'kaira.shah@example.com',
'female'),
('c64', 'Kian', 'Nair', 'Jade Street', 'Hazaribagh', 825301, 'kian.nair@example.com', 'male'),
('c65', 'Kyara', 'Venkataraman', 'Emerald Avenue', 'Hospet', 583201,
'kyara.venkataraman@example.com', 'female'),
('c66', 'Mehul', 'Ghosh', 'Silver Street', 'Hosur', 635109, 'mehul.ghosh@example.com', 'male'),
('c67', 'Miraya', 'Rao', 'Gold Avenue', 'Hubli', 580020, 'miraya.rao@example.com', 'female'),
('c68', 'Naksh', 'Saxena', 'Ruby Street', 'Kadapa', 516004, 'naksh.saxena@example.com',
'male'),
('c69', 'Navya', 'Kumar', 'Topaz Avenue', 'Kakinada', 533001, 'navya.kumar@example.com',

```

```

'female'),
('c70', 'Reyansh', 'Sharma', 'Sapphire Street', 'Kamarhati', 700058,
'reyansh.sharma@example.com', 'male'),
('c71', 'Rishika', 'Malhotra', 'Platinum Avenue', 'Kamarhati', 700058,
'rishika.malhotra@example.com', 'female'),
('c72', 'Rudra', 'Deshpande', 'Diamond Lane', 'Kamarhati', 700058,
'rudra.deshpande@example.com', 'male'),
('c73', 'Saisha', 'Nair', 'Coral Street', 'Kamarhati', 700058, 'saisha.nair@example.com', 'female'),
('c74', 'Samaira', 'Venkatesh', 'Amethyst Avenue', 'Kamarhati', 700058,
'samaira.venkatesh@example.com', 'female'),
('c75', 'Samar', 'Chowdhury', 'Jade Street', 'Kamarhati', 700058,
'samar.chowdhury@example.com', 'male'),
('c76', 'Sana', 'Sethi', 'Emerald Avenue', 'Kamarhati', 700058, 'sana.sethi@example.com',
'female'),
('c77', 'Sanskriti', 'Shah', 'Silver Street', 'Kamarhati', 700058, 'sanskriti.shah@example.com',
'female'),
('c78', 'Sara', 'Gupta', 'Gold Avenue', 'Kamarhati', 700058, 'sara.gupta@example.com', 'female'),
('c79', 'Satvik', 'Rao', 'Ruby Street', 'Kamarhati', 700058, 'satvik.rao@example.com', 'male'),
('c80', 'Shaan', 'Saxena', 'Topaz Avenue', 'Kamarhati', 700058, 'shaan.saxena@example.com',
'male'),
('c81', 'Shlok', 'Kumar', 'Sapphire Street', 'Kamarhati', 700058, 'shlok.kumar@example.com',
'male'),
('c82', 'Vihaan', 'Sharma', 'Platinum Avenue', 'Kamarhati', 700058,
'veihaan.sharma@example.com', 'male'),
('c83', 'Vivaan', 'Malhotra', 'Diamond Lane', 'Kamarhati', 700058,
'vevaan.malhotra@example.com', 'male'),
('c84', 'Yashvi', 'Deshpande', 'Coral Street', 'Kamarhati', 700058,
'yashvi.deshpande@example.com', 'female'),
('c85', 'Yuvaan', 'Nair', 'Amethyst Avenue', 'Kamarhati', 700058, 'yuvaan.nair@example.com',
'male'),
('c86', 'Advait', 'Venkataraman', 'Jade Street', 'Kanchipuram', 631501,
'advait.venkataraman@example.com', 'male'),
('c87', 'Ahaan', 'Ghosh', 'Emerald Avenue', 'Kanchipuram', 631501,
'ahaan.ghosh@example.com', 'male'),
('c88', 'Arnav', 'Rao', 'Silver Street', 'Kanchipuram', 631501, 'arnav.rao@example.com', 'male'),
('c89', 'Arya', 'Saxena', 'Gold Avenue', 'Kanchipuram', 631501, 'arya.saxena@example.com',
'female'),
('c90', 'Atharv', 'Kumar', 'Ruby Street', 'Kanchipuram', 631501, 'atharv.kumar@example.com',
'male');
-----.
-----.

```

INSERT INTO "Homecare".Client_Contact ("Client_Id", "Phone_Number") VALUES

('c1', 9876543210),
('c2', 9876543211),
('c3', 9876543212),
('c4', 9876543213),
('c5', 9876543214),
('c6', 9876543215),
('c7', 9876543216),
('c8', 9876543217),
('c9', 9876543218),
('c10', 9876543219),
('c11', 9876543220),
('c12', 9876543221),
('c13', 9876543222),
('c14', 9876543223),
('c15', 9876543224),
('c16', 9876543225),
('c17', 9876543226),
('c18', 9876543227),
('c19', 9876543228),
('c20', 9876543229),
('c21', 9876543230),
('c22', 9876543231),
('c23', 9876543232),
('c24', 9876543233),
('c25', 9876543234),
('c26', 9876543235),
('c27', 9876543236),
('c28', 9876543237),
('c29', 9876543238),
('c30', 9876543239),
('c31', 9876543240),
('c32', 9876543241),
('c33', 9876543242),
('c34', 9876543243),
('c35', 9876543244),
('c36', 9876543245),
('c37', 9876543246),
('c38', 9876543247),
('c39', 9876543248),
('c40', 9876543249),
('c41', 9876543250),
('c42', 9876543251),
('c43', 9876543252),
('c44', 9876543253),

('c45', 9876543254),
('c46', 9876543255),
('c47', 9876543256),
('c48', 9876543257),
('c49', 9876543258),
('c50', 9876543259),
('c51', 9876543260),
('c52', 9876543261),
('c53', 9876543262),
('c54', 9876543263),
('c55', 9876543264),
('c56', 9876543265),
('c57', 9876543266),
('c58', 9876543267),
('c59', 9876543268),
('c60', 9876543269),
('c61', 9876543270),
('c62', 9876543271),
('c63', 9876543272),
('c64', 9876543273),
('c65', 9876543274),
('c66', 9876543275),
('c67', 9876543276),
('c68', 9876543277),
('c69', 9876543278),
('c70', 9876543279),
('c71', 9876543280),
('c72', 9876543281),
('c73', 9876543282),
('c74', 9876543283),
('c75', 9876543284),
('c76', 9876543285),
('c77', 9876543286),
('c78', 9876543287),
('c79', 9876543288),
('c80', 9876543289),
('c81', 9876543290),
('c82', 9876543291),
('c83', 9876543292),
('c84', 9876543293),
('c85', 9876543294),
('c86', 9876543295),
('c87', 9876543296),
('c88', 9876543297),

('c89', 9876543298),
('c90', 9876543299);

INSERT INTO "Homecare".Caregivers ("Caregiver_Id", "First_Name", "Last_Name", "Dob", "Gender", "Email") VALUES
('car1', 'Rahul', 'Sharma', '1980-05-12', 'male', 'rahul.sharma@example.com'),
('car2', 'Priya', 'Gupta', '1982-07-23', 'female', 'priya.gupta@example.com'),
('car3', 'Amit', 'Verma', '1981-12-15', 'male', 'amit.verma@example.com'),
('car4', 'Sonia', 'Singh', '1985-09-30', 'female', 'sonia.singh@example.com'),
('car5', 'Vikram', 'Patel', '1979-04-19', 'male', 'vikram.patel@example.com'),
('car6', 'Anita', 'Rao', '1983-03-25', 'female', 'anita.rao@example.com'),
('car7', 'Raj', 'Malhotra', '1980-11-09', 'male', 'raj.malhotra@example.com'),
('car8', 'Meera', 'Kaur', '1984-08-14', 'female', 'meera.kaur@example.com'),
('car9', 'Rohan', 'Chopra', '1982-06-27', 'male', 'rohan.chopra@example.com'),
('car10', 'Sneha', 'Deshpande', '1986-02-08', 'female', 'sneha.deshpande@example.com'),
('car11', 'Sandeep', 'Bose', '1981-01-21', 'male', 'sandeep.bose@example.com'),
('car12', 'Neha', 'Reddy', '1983-10-17', 'female', 'neha.reddy@example.com'),
('car13', 'Aryan', 'Joshi', '1985-11-04', 'male', 'aryan.joshi@example.com'),
('car14', 'Pooja', 'Shah', '1982-09-09', 'female', 'pooja.shah@example.com'),
('car15', 'Aakash', 'Gandhi', '1980-12-31', 'male', 'aakash.gandhi@example.com'),
('car16', 'Anjali', 'Srinivasan', '1984-07-28', 'female', 'anjali.srinivasan@example.com'),
('car17', 'Alok', 'Iyer', '1981-06-18', 'male', 'alok.iyer@example.com'),
('car18', 'Smita', 'Chatterjee', '1983-02-14', 'female', 'smita.chatterjee@example.com'),
('car19', 'Rajesh', 'Venkatesh', '1985-04-02', 'male', 'rajesh.venkatesh@example.com'),
('car20', 'Kavita', 'Rao', '1980-03-09', 'female', 'kavita.rao@example.com'),
('car21', 'Amita', 'Saxena', '1982-08-07', 'female', 'amita.saxena@example.com'),
('car22', 'Vivek', 'Dixit', '1981-07-16', 'male', 'vivek.dixit@example.com'),
('car23', 'Nisha', 'Nair', '1984-11-28', 'female', 'nisha.nair@example.com'),
('car24', 'Sameer', 'Sharma', '1980-09-11', 'male', 'sameer.sharma@example.com'),
('car25', 'Kiran', 'Rastogi', '1983-06-29', 'female', 'kiran.rastogi@example.com'),
('car26', 'Vidya', 'Sethi', '1985-01-19', 'female', 'vidya.sethi@example.com'),
('car27', 'Rajat', 'Gupta', '1981-05-22', 'male', 'rajat.gupta@example.com'),
('car28', 'Arun', 'Shah', '1982-12-03', 'male', 'arun.shah@example.com'),
('car29', 'Sunita', 'Malhotra', '1984-10-08', 'female', 'sunita.malhotra@example.com'),
('car30', 'Sanjay', 'Deshpande', '1980-11-14', 'male', 'sanjay.deshpande@example.com'),
('car31', 'Ananya', 'Bose', '1983-02-07', 'female', 'ananya.bose@example.com'),
('car32', 'Arjun', 'Reddy', '1985-03-28', 'male', 'arjun.reddy@example.com'),
('car33', 'Kajal', 'Joshi', '1981-08-09', 'female', 'kajal.joshi@example.com'),
('car34', 'Amit', 'Shah', '1984-07-19', 'male', 'amit.shah@example.com'),
('car35', 'Riya', 'Gandhi', '1980-06-22', 'female', 'riya.gandhi@example.com'),
('car36', 'Alok', 'Srinivasan', '1982-01-04', 'male', 'alok.srinivasan@example.com'),

```

('car37', 'Sarita', 'Iyer', '1981-10-13', 'female', 'sarita.iyer@example.com'),
('car38', 'Vikas', 'Chatterjee', '1983-12-27', 'male', 'vikas.chatterjee@example.com'),
('car39', 'Seema', 'Venkatesh', '1985-09-30', 'female', 'seema.venkatesh@example.com'),
('car40', 'Ashok', 'Rao', '1980-07-11', 'male', 'ashok.rao@example.com'),
('car41', 'Neha', 'Saxena', '1982-02-12', 'female', 'neha.saxena@example.com'),
('car42', 'Suresh', 'Dixit', '1981-01-26', 'male', 'suresh.dixit@example.com'),
('car43', 'Rani', 'Nair', '1983-08-18', 'female', 'rani.nair@example.com'),
('car44', 'Rahul', 'Sharma', '1985-06-05', 'male', 'rahul.sharma@example.com'),
('car45', 'Priya', 'Gupta', '1980-09-09', 'female', 'priya.gupta@example.com'),
('car46', 'Amit', 'Verma', '1982-12-24', 'male', 'amit.verma@example.com'),
('car47', 'Sonia', 'Singh', '1981-11-03', 'female', 'sonia.singh@example.com'),
('car48', 'Vikram', 'Patel', '1983-05-14', 'male', 'vikram.patel@example.com'),
('car49', 'Anita', 'Rao', '1985-04-07', 'female', 'anita.rao@example.com'),
('car50', 'Raj', 'Malhotra', '1980-03-19', 'male', 'raj.malhotra@example.com');

```

```

INSERT INTO "Homecare".Caregivers ("Caregiver_Id", "First_Name", "Last_Name", "Dob",
"Gender", "Email") VALUES
('car51', 'Vivek', 'Shah', '1982-05-12', 'male', 'vivek.shah@example.com'),
('car52', 'Kavita', 'Malhotra', '1984-07-23', 'female', 'kavita.malhotra@example.com'),
('car53', 'Alok', 'Verma', '1981-12-15', 'male', 'alok.verma@example.com'),
('car54', 'Sneha', 'Singh', '1985-09-30', 'female', 'sneha.singh@example.com'),
('car55', 'Surya', 'Patel', '1979-04-19', 'male', 'surya.patel@example.com'),
('car56', 'Nisha', 'Rao', '1983-03-25', 'female', 'nisha.rao@example.com'),
('car57', 'Rajat', 'Malhotra', '1980-11-09', 'male', 'rajat.malhotra@example.com'),
('car58', 'Anjali', 'Kaur', '1984-08-14', 'female', 'anjali.kaur@example.com'),
('car59', 'Rohit', 'Chopra', '1982-06-27', 'male', 'rohit.chopra@example.com'),
('car60', 'Seema', 'Deshpande', '1986-02-08', 'female', 'seema.deshpande@example.com'),
('car61', 'Sanjay', 'Bose', '1981-01-21', 'male', 'sanjay.bose@example.com'),
('car62', 'Ananya', 'Reddy', '1983-10-17', 'female', 'ananya.reddy@example.com'),
('car63', 'Amit', 'Joshi', '1985-11-04', 'male', 'amit.joshi@example.com'),
('car64', 'Poonam', 'Shah', '1982-09-09', 'female', 'poonam.shah@example.com'),
('car65', 'Aaditya', 'Gandhi', '1980-12-31', 'male', 'aaditya.gandhi@example.com'),
('car66', 'Aisha', 'Srinivasan', '1984-07-28', 'female', 'aisha.srinivasan@example.com'),
('car67', 'Akash', 'Iyer', '1981-06-18', 'male', 'akash.iyer@example.com'),
('car68', 'Sunita', 'Chatterjee', '1983-02-14', 'female', 'sunita.chatterjee@example.com'),
('car69', 'Rakesh', 'Venkatesh', '1985-04-02', 'male', 'rakesh.venkatesh@example.com'),
('car70', 'Kiran', 'Rao', '1980-03-09', 'female', 'kiran.rao@example.com'),
('car71', 'Amita', 'Saxena', '1982-08-07', 'female', 'amita.saxena@example.com'),
('car72', 'Vikas', 'Dixit', '1981-07-16', 'male', 'vikas.dixit@example.com'),
('car73', 'Neha', 'Nair', '1984-11-28', 'female', 'neha.nair@example.com'),
('car74', 'Sameer', 'Sharma', '1980-09-11', 'male', 'sameer.sharma@example.com'),
('car75', 'Kavita', 'Rastogi', '1983-06-29', 'female', 'kavita.rastogi@example.com'),
('car76', 'Vijay', 'Sethi', '1985-01-19', 'male', 'vijay.sethi@example.com'),
('car77', 'Rajat', 'Gupta', '1981-05-22', 'male', 'rajat.gupta@example.com'),

```

```
('car78', 'Arpita', 'Shah', '1982-12-03', 'female', 'arpita.shah@example.com'),  
('car79', 'Sarita', 'Malhotra', '1984-10-08', 'female', 'sarita.malhotra@example.com'),  
('car80', 'Sanjay', 'Deshpande', '1980-11-14', 'male', 'sanjay.deshpande@example.com'),  
('car81', 'Ananya', 'Bose', '1983-02-07', 'female', 'ananya.bose@example.com'),  
('car82', 'Arjun', 'Reddy', '1985-03-28', 'male', 'arjun.reddy@example.com'),  
('car83', 'Kajal', 'Joshi', '1981-08-09', 'female', 'kajal.joshi@example.com'),  
('car84', 'Amit', 'Shah', '1984-07-19', 'male', 'amit.shah@example.com'),  
('car85', 'Riya', 'Gandhi', '1980-06-22', 'female', 'riya.gandhi@example.com'),  
('car86', 'Alok', 'Srinivasan', '1982-01-04', 'male', 'alok.srinivasan@example.com'),  
('car87', 'Sarita', 'Iyer', '1981-10-13', 'female', 'sarita.iyer@example.com'),  
('car88', 'Vikas', 'Chatterjee', '1983-12-27', 'male', 'vikas.chatterjee@example.com'),  
('car89', 'Seema', 'Venkatesh', '1985-09-30', 'female', 'seema.venkatesh@example.com'),  
('car90', 'Ashok', 'Rao', '1980-07-11', 'male', 'ashok.rao@example.com');
```

```
INSERT INTO "Homecare".caregiver_Contact ("Caregiver_Id", "Phone_Number") VALUES  
('car1', 1111111111),  
('car2', 2222222222),  
('car3', 3333333333),  
('car4', 4444444444),  
('car5', 5555555555),  
('car6', 6666666666),  
('car7', 7777777777),  
('car8', 8888888888),  
('car9', 9999999999),  
('car10', 1111111111), --  
('car11', 2222222222), --  
('car12', 3333333333), --  
('car13', 4444444444), --  
('car14', 5555555555), --  
('car15', 6666666666), --  
('car16', 7777777777), --  
('car17', 8888888888),  
('car18', 9999999999),  
('car19', 1234567890),  
('car20', 2345678901),  
('car21', 3456789012),  
('car22', 4567890123),  
('car23', 5678901234),  
('car24', 6789012345),  
('car25', 7890123456),  
('car26', 8901234567),
```

```
('car27', 9012345678),
('car28', 9876543210),
('car29', 8765432109),
('car30', 7654321098),
('car31', 6543210987),
('car32', 5432109876),
('car33', 4321098765),
('car34', 3210987654),
('car35', 2109876543),
('car36', 1098765432),
('car37', 1098765432), --
('car38', 1098765432), --
('car39', 1098765432), --
('car40', 1098765432), --
('car41', 1111222233),
('car42', 1122334455),
('car43', 1234123412),
('car44', 5555666677),
('car45', 1212121212),
('car46', 2121212121);
```

```
INSERT INTO "Homecare".caregiver_Contact ("Caregiver_Id", "Phone_Number") VALUES
('car47', 7777888899),
('car48', 8888999900),
('car49', 9999000011),
('car50', 1234432156),
('car51', 6543212345),
('car52', 9876987698),
('car53', 1213141516),
('car54', 1615141312),
('car55', 3333444455),
('car56', 4555666777),
('car57', 2345234523),
('car58', 9876987698),
('car59', 7654765476),
('car60', 2345234523),
('car61', 3456345634),
('car62', 4567456745),
('car63', 5678567856),
('car64', 6789678967),
('car65', 7890789078),
('car66', 8901890189),
('car67', 9012901290),
('car68', 1010101010),
```

```
('car69', 2020202020),  
('car70', 3030303030),  
('car71', 4040404040),  
('car72', 5050505050),  
('car73', 6060606060),  
('car74', 7070707070),  
('car75', 8080808080),  
('car76', 9090909090),  
('car77', 1122334455),  
('car78', 4455667788),  
('car79', 1133557799),  
('car80', 2244668800),  
('car81', 3366990011),  
('car82', 4488221133),  
('car83', 5500552277),  
('car84', 6611663388),  
('car85', 7722774499),  
('car86', 8833885611),  
('car87', 9944996722),  
('car88', 1234567890),  
('car89', 2345678901),  
('car90', 3456789012);
```

```
INSERT INTO "Homecare".Service ("Service_Id", "Caregiver_Id", "Charges", "Service_Type")  
VALUES  
('s1', 'car1', 5000, 'Physiotherapy'),  
('s2', 'car2', 3000, 'Nursing'),  
('s3', 'car3', 2500, 'Elderly Care'),  
('s4', 'car4', 4000, 'Physical Therapy'),  
('s5', 'car5', 3500, 'Nursing'),  
('s6', 'car6', 2000, 'Elderly Care'),  
('s7', 'car7', 4500, 'Physiotherapy'),  
('s8', 'car8', 4000, 'Physical Therapy'),  
('s9', 'car9', 3000, 'Nursing'),  
('s10', 'car10', 2500, 'Elderly Care'),  
('s11', 'car11', 3500, 'Nursing'),  
('s12', 'car12', 3000, 'Elderly Care'),  
('s13', 'car13', 4000, 'Physiotherapy'),  
('s14', 'car14', 4500, 'Physical Therapy'),  
('s15', 'car15', 2000, 'Nursing'),  
('s16', 'car16', 5000, 'Elderly Care'),
```

```

('s17', 'car17', 2500, 'Physiotherapy'),
('s18', 'car18', 3000, 'Physical Therapy'),
('s19', 'car19', 3500, 'Nursing'),
('s20', 'car20', 4000, 'Elderly Care'),
('s21', 'car21', 3000, 'Nursing'),
('s22', 'car22', 2500, 'Elderly Care'),
('s23', 'car23', 4000, 'Physiotherapy'),
('s24', 'car24', 4500, 'Physical Therapy'),
('s25', 'car25', 2000, 'Nursing'),
('s26', 'car26', 5000, 'Elderly Care'),
('s27', 'car27', 2500, 'Physiotherapy'),
('s28', 'car28', 3000, 'Physical Therapy'),
('s29', 'car29', 3500, 'Nursing'),
('s30', 'car30', 4000, 'Elderly Care'),
('s31', 'car31', 3000, 'Nursing'),
('s32', 'car32', 2500, 'Elderly Care'),
('s33', 'car33', 4000, 'Physiotherapy'),
('s34', 'car34', 4500, 'Physical Therapy'),
('s35', 'car35', 2000, 'Nursing'),
('s36', 'car36', 5000, 'Elderly Care'),
('s37', 'car37', 2500, 'Physiotherapy'),
('s38', 'car38', 3000, 'Physical Therapy'),
('s39', 'car39', 3500, 'Nursing'),
('s40', 'car40', 4000, 'Elderly Care'),
('s41', 'car41', 3000, 'Nursing'),
('s42', 'car42', 2500, 'Elderly Care'),
('s43', 'car43', 4000, 'Physiotherapy'),
('s44', 'car44', 4500, 'Physical Therapy'),
('s45', 'car45', 2000, 'Nursing'),
('s46', 'car46', 5000, 'Elderly Care'),
('s47', 'car47', 2500, 'Physiotherapy'),
('s48', 'car48', 3000, 'Physical Therapy'),
('s49', 'car49', 3500, 'Nursing'),
('s50', 'car50', 4000, 'Elderly Care');

```

```

INSERT INTO "Homecare".Service ("Service_Id", "Caregiver_Id", "Charges", "Service_Type")
VALUES
('s51', 'car1', 3500, 'Nursing'),
('s52', 'car2', 3000, 'Elderly Care'),
('s53', 'car2', 4000, 'Physiotherapy'),
('s54', 'car3', 4500, 'Physical Therapy'),
('s55', 'car3', 2500, 'Nursing'),
('s56', 'car4', 3500, 'Elderly Care'),
('s57', 'car5', 3000, 'Elderly Care'),

```

```
('s58', 'car5', 4000, 'Physiotherapy'),  
('s59', 'car6', 2500, 'Physical Therapy'),  
('s60', 'car6', 4500, 'Nursing'),  
('s61', 'car7', 3500, 'Nursing'),  
('s62', 'car8', 3000, 'Elderly Care'),  
('s63', 'car9', 4000, 'Physiotherapy'),  
('s64', 'car9', 2500, 'Physical Therapy'),  
('s65', 'car10', 4500, 'Nursing'),  
('s66', 'car10', 3500, 'Nursing'),  
('s67', 'car11', 3000, 'Elderly Care'),  
('s68', 'car12', 4000, 'Physiotherapy'),  
('s69', 'car13', 2500, 'Physical Therapy'),  
('s70', 'car14', 4500, 'Nursing'),  
('s71', 'car14', 3500, 'Nursing'),  
('s72', 'car15', 3000, 'Elderly Care'),  
('s73', 'car16', 4000, 'Physiotherapy'),  
('s74', 'car16', 2500, 'Physical Therapy'),  
('s75', 'car17', 4500, 'Nursing'),  
('s76', 'car17', 3500, 'Nursing'),  
('s77', 'car18', 3000, 'Elderly Care'),  
('s78', 'car19', 4000, 'Physiotherapy'),  
('s79', 'car19', 2500, 'Physical Therapy'),  
('s80', 'car20', 4500, 'Nursing');
```

```
INSERT INTO "Homecare".Appointments ("Appointment_Id", "Service_Id", "Client_Id", "Date",  
"Status") VALUES  
('a1', 's1', 'c1', '2023-12-05', 'pending'),  
('a2', 's2', 'c2', '2023-12-06', 'pending'),  
('a3', 's3', 'c3', '2023-12-07', 'complete'),  
('a4', 's4', 'c4', '2023-12-08', 'pending'),  
('a5', 's5', 'c5', '2023-12-09', 'complete'),  
('a6', 's6', 'c6', '2023-12-10', 'pending'),  
('a7', 's7', 'c7', '2023-12-11', 'pending'),  
('a8', 's8', 'c8', '2023-12-12', 'complete'),  
('a9', 's9', 'c9', '2023-12-13', 'pending'),  
('a10', 's10', 'c10', '2023-12-14', 'complete'),  
('a11', 's11', 'c11', '2023-12-15', 'pending'),  
('a12', 's12', 'c12', '2023-12-16', 'pending'),  
('a13', 's13', 'c13', '2023-12-17', 'complete'),  
('a14', 's14', 'c14', '2023-12-18', 'pending'),
```

```

('a15', 's15', 'c15', '2023-12-19', 'complete'),
('a16', 's16', 'c16', '2023-12-20', 'pending'),
('a17', 's17', 'c17', '2023-12-21', 'pending'),
('a18', 's18', 'c18', '2023-12-22', 'complete'),
('a19', 's19', 'c19', '2023-12-23', 'pending'),
('a20', 's20', 'c20', '2023-12-24', 'complete'),
('a21', 's21', 'c21', '2023-12-25', 'pending'),
('a22', 's22', 'c22', '2023-12-26', 'pending'),
('a23', 's23', 'c23', '2023-12-27', 'complete'),
('a24', 's24', 'c24', '2023-12-28', 'pending'),
('a25', 's25', 'c25', '2023-12-29', 'complete'),
('a26', 's26', 'c26', '2023-12-30', 'pending'),
('a27', 's27', 'c27', '2023-12-31', 'pending'),
('a28', 's28', 'c28', '2024-01-01', 'complete'),
('a29', 's29', 'c29', '2024-01-02', 'pending'),
('a30', 's30', 'c30', '2024-01-03', 'complete'),
('a31', 's31', 'c31', '2024-01-04', 'pending'),
('a32', 's32', 'c32', '2024-01-05', 'pending'),
('a33', 's33', 'c33', '2024-01-06', 'complete'),
('a34', 's34', 'c34', '2024-01-07', 'pending'),
('a35', 's35', 'c35', '2024-01-08', 'complete'),
('a36', 's36', 'c36', '2024-01-09', 'pending'),
('a37', 's37', 'c37', '2024-01-10', 'pending'),
('a38', 's38', 'c38', '2024-01-11', 'complete'),
('a39', 's39', 'c39', '2024-01-12', 'pending'),
('a40', 's40', 'c40', '2024-01-13', 'complete');

```

```

INSERT INTO "Homecare".Appointments ("Appointment_Id", "Service_Id", "Client_Id", "Date",
>Status") VALUES
('a41', 's1', 'c10', '2024-01-14', 'rejected'),
('a42', 's2', 'c5', '2024-01-15', 'complete'),
('a43', 's2', 'c6', '2024-01-16', 'rejected'),
('a44', 's3', 'c6', '2024-01-17', 'pending'),
('a45', 's4', 'c7', '2024-01-18', 'rejected'),
('a46', 's5', 'c3', '2024-01-19', 'complete'),
('a47', 's6', 'c4', '2024-01-20', 'rejected'),
('a48', 's7', 'c7', '2024-01-21', 'pending'),
('a49', 's8', 'c8', '2024-01-22', 'rejected'),
('a50', 's9', 'c9', '2024-01-23', 'complete'),
('a51', 's10', 'c10', '2024-01-24', 'rejected'),
('a52', 's11', 'c11', '2024-01-25', 'pending'),
('a53', 's12', 'c12', '2024-01-26', 'rejected'),
('a54', 's13', 'c13', '2024-01-27', 'complete'),
('a55', 's14', 'c14', '2024-01-28', 'rejected'),

```

```
('a56', 's15', 'c15', '2024-01-29', 'pending'),  
('a57', 's16', 'c16', '2024-01-30', 'rejected'),  
('a58', 's17', 'c17', '2024-01-31', 'complete'),  
('a59', 's18', 'c18', '2024-02-01', 'rejected'),  
('a60', 's19', 'c19', '2024-02-02', 'pending'),  
('a61', 's20', 'c20', '2024-02-03', 'rejected'),  
('a62', 's21', 'c21', '2024-02-04', 'complete'),  
('a63', 's22', 'c22', '2024-02-05', 'rejected'),  
('a64', 's23', 'c23', '2024-02-06', 'pending'),  
('a65', 's24', 'c24', '2024-02-07', 'rejected'),  
('a66', 's25', 'c25', '2024-02-08', 'complete'),  
('a67', 's26', 'c26', '2024-02-09', 'rejected'),  
('a68', 's27', 'c27', '2024-02-10', 'pending'),  
('a69', 's28', 'c28', '2024-02-11', 'rejected'),  
('a70', 's29', 'c29', '2024-02-12', 'complete'),  
('a71', 's30', 'c30', '2024-02-13', 'rejected'),  
('a72', 's31', 'c31', '2024-02-14', 'pending'),  
('a73', 's32', 'c32', '2024-02-15', 'rejected'),  
('a74', 's33', 'c33', '2024-02-16', 'complete'),  
('a75', 's34', 'c34', '2024-02-17', 'rejected'),  
('a76', 's35', 'c35', '2024-02-18', 'pending'),  
('a77', 's36', 'c36', '2024-02-19', 'rejected'),  
('a78', 's37', 'c37', '2024-02-20', 'complete'),  
('a79', 's38', 'c38', '2024-02-21', 'rejected'),  
('a80', 's39', 'c39', '2024-02-22', 'pending');
```

INSERT INTO "Homecare".Appointments ("Appointment_Id", "Service_Id", "Client_Id", "Date", "Status") VALUES

```
('a41', 's1', 'c41', '2024-01-14', 'rejected'),  
('a42', 's2', 'c42', '2024-01-15', 'rejected'),  
('a43', 's2', 'c43', '2024-01-16', 'pending'),  
('a44', 's3', 'c44', '2024-01-17', 'rejected'),  
('a45', 's3', 'c45', '2024-01-18', 'complete'),  
('a46', 's4', 'c46', '2024-01-19', 'rejected'),  
('a47', 's4', 'c47', '2024-01-20', 'pending'),  
('a48', 's5', 'c48', '2024-01-21', 'rejected'),  
('a49', 's5', 'c49', '2024-01-22', 'complete'),  
('a50', 's6', 'c50', '2024-01-23', 'rejected'),  
('a51', 's6', 'c51', '2024-01-24', 'pending'),  
('a52', 's7', 'c52', '2024-01-25', 'rejected'),  
('a53', 's7', 'c53', '2024-01-26', 'complete'),  
('a54', 's8', 'c54', '2024-01-27', 'rejected'),  
('a55', 's8', 'c55', '2024-01-28', 'pending'),  
('a56', 's9', 'c56', '2024-01-29', 'rejected'),
```

```
('a57', 's9', 'c57', '2024-01-30', 'complete'),  
('a58', 's10', 'c58', '2024-01-31', 'rejected'),  
('a59', 's10', 'c59', '2024-02-01', 'pending'),  
('a60', 's11', 'c60', '2024-02-02', 'rejected'),  
('a61', 's11', 'c61', '2024-02-03', 'complete'),  
('a62', 's12', 'c62', '2024-02-04', 'rejected'),  
('a63', 's12', 'c63', '2024-02-05', 'pending'),  
('a64', 's13', 'c64', '2024-02-06', 'rejected'),  
('a65', 's13', 'c65', '2024-02-07', 'complete'),  
('a66', 's14', 'c66', '2024-02-08', 'rejected'),  
('a67', 's14', 'c67', '2024-02-09', 'pending'),  
('a68', 's15', 'c68', '2024-02-10', 'rejected'),  
('a69', 's15', 'c69', '2024-02-11', 'complete'),  
('a70', 's16', 'c70', '2024-02-12', 'rejected'),  
('a71', 's16', 'c71', '2024-02-13', 'pending'),  
('a72', 's17', 'c72', '2024-02-14', 'rejected'),  
('a73', 's17', 'c73', '2024-02-15', 'complete'),  
('a74', 's18', 'c74', '2024-02-16', 'rejected'),  
('a75', 's18', 'c75', '2024-02-17', 'pending'),  
('a76', 's19', 'c76', '2024-02-18', 'rejected'),  
('a77', 's19', 'c77', '2024-02-19', 'complete'),  
('a78', 's20', 'c78', '2024-02-20', 'rejected'),  
('a79', 's20', 'c79', '2024-02-21', 'pending');
```

```
INSERT INTO "Homecare".bill ("Apppointment_id", "Total_payment", "Bill_Date",  
"Payment_status") VALUES  
('a3', 5000, '2024-01-01', 'done'),  
('a5', 4000, '2024-01-02', 'pending'),  
('a8', 4500, '2024-01-03', 'done'),  
('a10', 3000, '2024-01-04', 'pending'),  
('a13', 3500, '2024-01-05', 'done'),  
('a15', 2000, '2024-01-06', 'pending'),  
('a18', 2500, '2024-01-07', 'done'),  
('a20', 4500, '2024-01-08', 'done'),  
('a23', 4000, '2024-01-09', 'pending'),  
('a25', 3000, '2024-01-10', 'done'),  
('a28', 3500, '2024-01-11', 'pending'),  
('a30', 5000, '2024-01-12', 'done'),  
('a33', 2000, '2024-01-13', 'done'),
```

```
('a35', 2500, '2024-01-14', 'pending'),  
('a38', 4500, '2024-01-15', 'done'),  
('a40', 4000, '2024-01-16', 'done'),  
('a45', 3000, '2024-01-17', 'pending'),  
('a49', 3500, '2024-01-18', 'done'),  
('a53', 5000, '2024-01-19', 'pending'),  
('a57', 2000, '2024-01-20', 'done'),  
('a61', 2500, '2024-01-21', 'done'),  
('a65', 4500, '2024-01-22', 'pending'),  
('a69', 4000, '2024-01-23', 'done'),  
('a73', 3000, '2024-01-24', 'done'),  
('a77', 3500, '2024-01-25', 'pending');
```

```
INSERT INTO "Homecare".Feedback ("Appointment_id", "Date", "Rating", "description")  
VALUES  
('a3', '2024-01-01', 5, 'Excellent service.'),  
('a5', '2024-01-02', 4, 'The caregiver was very polite and professional.'),  
('a8', '2024-01-03', 5, 'Extremely satisfied with the service.'),  
('a10', '2024-01-04', 3, 'Satisfactory, but room for improvement.'),  
('a13', '2024-01-05', 4, 'The caregiver was punctual and attentive.'),  
('a15', '2024-01-06', 2, 'Not very happy with the service.'),  
('a18', '2024-01-07', 3, 'Average service.'),  
('a20', '2024-01-08', 5, 'Highly recommended.'),  
('a23', '2024-01-09', 4, 'Good service overall.'),  
('a25', '2024-01-10', 3, 'Could have been better.'),  
('a28', '2024-01-11', 4, 'Satisfied with the caregiver.'),  
('a30', '2024-01-12', 5, 'The caregiver was very helpful.'),  
('a33', '2024-01-13', 2, 'Disappointed with the service.'),  
('a35', '2024-01-14', 3, 'Okay, but could be improved.'),  
('a38', '2024-01-15', 4, 'Quite satisfied.'),  
('a40', '2024-01-16', 5, 'The caregiver was exceptional.'),  
('a45', '2024-01-17', 4, 'No major complaints.'),  
('a49', '2024-01-18', 3, 'Fairly good service.'),  
('a53', '2024-01-19', 2, 'Needs improvement.'),  
('a57', '2024-01-20', 4, 'The caregiver was friendly and understanding.'),  
('a51', '2024-01-21', 5, 'Very impressed with the service.'),  
('a65', '2024-01-22', 3, 'The service was satisfactory.'),  
('a69', '2024-01-23', 2, 'Not up to the mark.'),  
('a73', '2024-01-24', 4, 'The caregiver was knowledgeable and caring.'),
```

('a77', '2024-01-25', 5, 'Excellent care and attention provided.');

QUERY IN PLAIN ENGLISH

Basic Queries:

1. Retrieve all clients' information.
2. List all caregivers with their contact details.
3. Get a list of services along with their charges.
4. Find all appointments scheduled for a specific date.
5. List all clients who have pending appointments.
6. Retrieve the total payment of a specific bill.
7. Find the appointment with the highest total payment.
8. List all appointments with a "complete" status.
9. Find the caregiver with the most appointments.
10. Find the clients who provided a rating below 3 in feedback.
11. Get the total number of clients in the system.
12. List caregivers who have not received any feedback.
13. Find all bills with a "done" payment status.

Complex Queries:

1. Calculate the total earnings of a specific caregiver.
2. Find the top 5 caregivers with the highest earnings.
3. Identify clients who have appointments on multiple dates.
4. List clients who haven't scheduled any appointments.
5. Find the service types provided by a specific caregiver.
6. Calculate the total earnings for each caregiver and display it in descending order.
7. Retrieve the last appointment date for each client.
8. List caregivers who have provided services in more than one city.
9. Calculate the total revenue for the Homecare service.
10. Identify clients with the most appointments in the past month.
11. Find the appointments that have the same service and date.
12. List caregivers with a "female" gender who have appointments.
13. List clients who have pending appointments and their contact details.
14. Calculate the total number of completed appointments for each caregiver.
15. Identify the most popular service type based on the number of appointments.
16. List caregivers who have not been assigned any appointments.
17. Find the appointments scheduled for clients with a specific pin code.

18. Calculate the total payments for "done" bills in the last month.
19. List clients who provided feedback with a rating of 5.
20. Identify appointments that have a "complete" payment status but no feedback.
21. Find caregivers with "complete" appointments who have been rated poorly.
22. Calculate the average rating for each caregiver.
23. Identify the services provided on a specific date.
24. Calculate the total earnings for each service type.
25. List clients who have provided feedback and their ratings.
26. Identify the most profitable service based on the total payment.
27. Find the average total payment for appointments for each city.

Stored Procedures and Triggers:

1. Create a stored procedure to calculate a caregiver's total earnings.
2. Implement a stored procedure to calculate the average rating for each caregiver.
3. Implement a stored procedure to generate a monthly report of earnings for caregivers.
4. Implement a stored procedure to find and list clients with no scheduled appointments.
5. Create a trigger to automatically add a bill for completed appointments with a payment status of "pending."
6. Create a trigger to prevent the deletion of a caregiver record if they have pending appointments.

→ Retrieve all clients' information.

SELECT * FROM "Homecare".Client;

```
1 SELECT * FROM "Homecare".Client;
```

Data output Messages Notifications

	Client_Id [PK] character varying	First_Name character varying	Last_Name character varying	Street character varying	City character varying	Pincode integer	Email character varying	Gender character varying
1	c1	Aarav	Kumar	Surya Nagar	Mumbai	400001	aarav.kumar@exa...	male
2	c2	Aarya	Sharma	Lotus Colony	Delhi	110001	aarya.sharma@e...	female
3	c3	Aditi	Joshi	Rose Street	Bangalore	560001	aditi.joshi@exam...	female
4	c4	Amit	Patel	River View	Ahmedabad	380001	amit.patel@exa...	male
5	c5	Anaya	Singh	Golden Avenue	Chennai	600001	anaya.singh@ex...	female
6	c6	Arjun	Gupta	Heritage Road	Kolkata	700001	arjun.gupta@exa...	male
7	c7	Avni	Shah	Diamond Lane	Hyderabad	500001	avni.shah@exam...	female
8	c8	Dev	Verma	Silver Street	Pune	411001	dev.verma@exa...	male
9	c9	Esha	Rao	Jade Lane	Surat	395001	esha.rao@examp...	female
10	c10	Ishaan	Malhotra	Emerald Road	Jaipur	302001	ishaan.malhotra...	male
11	c11	Kabir	Deshpande	Topaz Lane	Lucknow	226001	kabir.despande...	male
12	c12	Kavya	Reddy	Amber Street	Nagpur	440001	kavya.reddy@exa...	female
13	c13	Leela	Gandhi	Sapphire Lane	Indore	452001	leela.gandhi@ex...	female
14	c14	Mira	Chatterjee	Platinum Avenue	Vadodara	390001	mira.chatterjee@...	female
15	c15	Neha	Bose	Pearl Street	Kanpur	208001	neha.bose@exa...	female
16	c16	Om	Dixit	Ruby Lane	Bhopal	462001	om.dixit@exampl...	male
17	c17	Pari	Sharma	Opal Street	Patna	800001	pari.sharma@exa...	female
18	c18	Pranav	Iyer	Coral Lane	Ludhiana	141001	pranav.iyer@exa...	male
19	c19	Riya	Chopra	Amethyst Road	Agra	282001	riya.chopra@exa...	female

Total rows: 90 of 90 Query complete 00:00:00.215

→List all caregivers with their contact details.

```
SELECT caregivers.*, caregiver_Contact."Phone_Number"
FROM "Homecare".caregivers
JOIN "Homecare".caregiver_Contact
ON caregivers."Caregiver_Id" = caregiver_Contact."Caregiver_Id";
```

Query Query History

```
1 SELECT caregivers.*, caregiver_Contact."Phone_Number"
2 FROM "Homecare".caregivers
3 JOIN "Homecare".caregiver_Contact
4 ON caregivers."Caregiver_Id" = caregiver_Contact."Caregiver_Id";
5
```

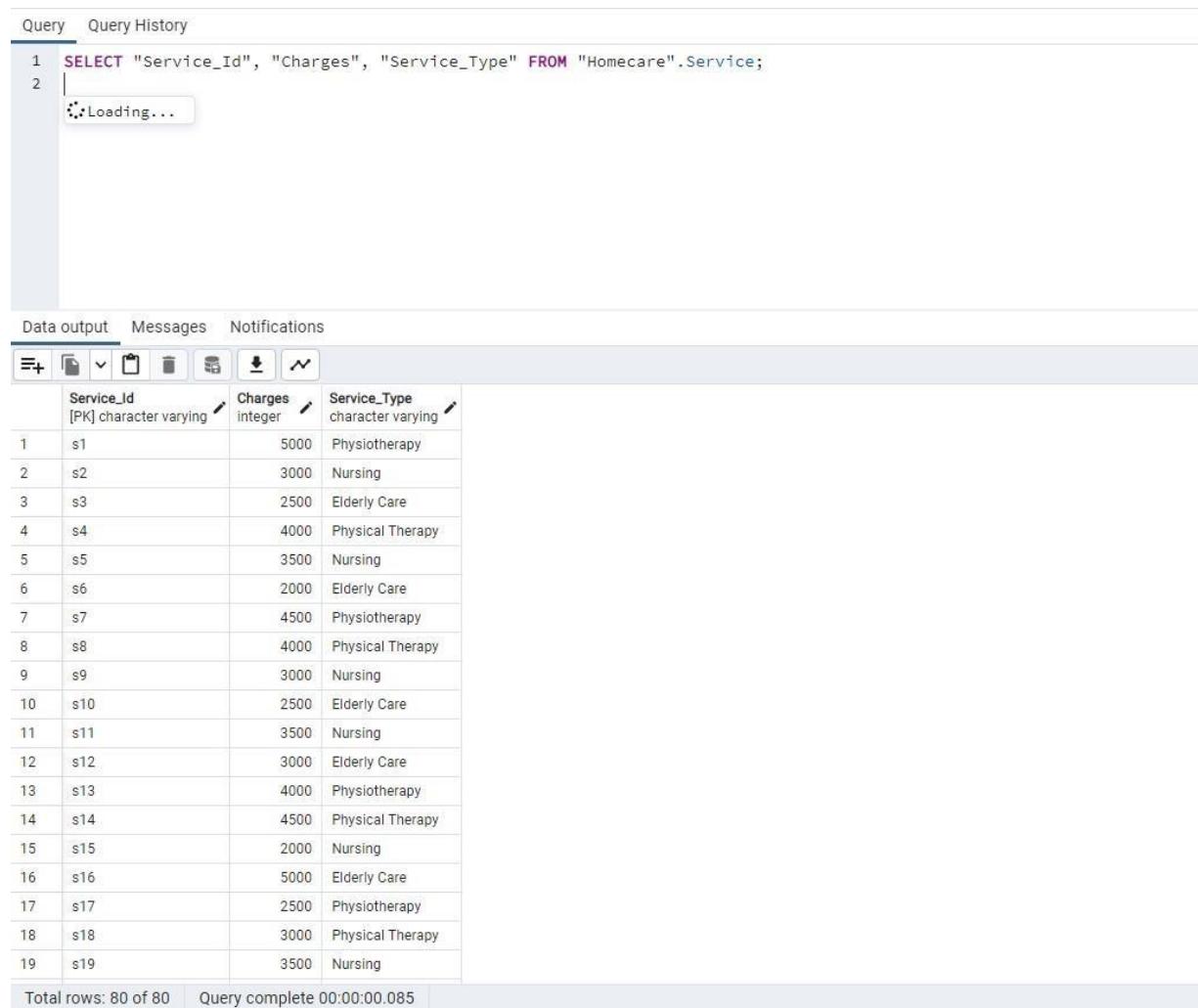
Data output Messages Notifications

	Caregiver_Id character varying	First_Name character varying	Last_Name character varying	Dob date	Gender character varying	Email character varying	Phone_Number numeric
1	car1	Rahul	Sharma	1980-05-12	male	rahul.sharma@example.com	1111111111
2	car2	Priya	Gupta	1982-07-23	female	priya.gupta@example.com	2222222222
3	car3	Amit	Verma	1981-12-15	male	amit.verma@example.com	3333333333
4	car4	Sonia	Singh	1985-09-30	female	sonia.singh@example.com	4444444444
5	car5	Vikram	Patel	1979-04-19	male	vikram.patel@example.com	5555555555
6	car6	Anita	Rao	1983-03-25	female	anita.rao@example.com	6666666666
7	car7	Raj	Malhotra	1980-11-09	male	raj.malhotra@example.com	7777777777
8	car8	Meera	Kaur	1984-08-14	female	meera.kaur@example.com	8888888888
9	car9	Rohan	Chopra	1982-06-27	male	rohan.chopra@example.com	9999999999
10	car10	Sneha	Deshpande	1986-02-08	female	sneha.deshpande@example.com	1111111111
11	car11	Sandeep	Bose	1981-01-21	male	sandeep.bose@example.com	2222222222
12	car12	Neha	Reddy	1983-10-17	female	neha.reddy@example.com	3333333333
13	car13	Aryan	Joshi	1985-11-04	male	aryan.joshi@example.com	4444444444
14	car14	Pooja	Shah	1982-09-09	female	pooja.shah@example.com	5555555555
15	car15	Aakash	Gandhi	1980-12-31	male	aakash.gandhi@example.com	6666666666
16	car16	Anjali	Srinivasan	1984-07-28	female	anjali.srinivasan@example.com	7777777777
17	car17	Alok	Iyer	1981-06-18	male	alok.iyer@example.com	8888888888
18	car18	Smita	Chatterjee	1983-02-14	female	smita.chatterjee@example.com	9999999999
19	car19	Rajesh	Venkatesh	1985-04-02	male	rajesh.venkatesh@example.com	1234567890

Total rows: 90 of 90 Query complete 00:00:00.052

→Get a list of services along with their charges.

SELECT "Service_Id", "Charges", "Service_Type" FROM "Homecare".Service;



The screenshot shows a database query results window. At the top, there are tabs for 'Query' (which is selected) and 'Query History'. Below the tabs, the SQL query is displayed:

```
1 SELECT "Service_Id", "Charges", "Service_Type" FROM "Homecare".Service;
2 |
```

A progress bar indicates the data is 'Loading...'. Once loaded, the results are presented in a table format:

	Service_Id [PK] character varying	Charges integer	Service_Type character varying
1	s1	5000	Physiotherapy
2	s2	3000	Nursing
3	s3	2500	Elderly Care
4	s4	4000	Physical Therapy
5	s5	3500	Nursing
6	s6	2000	Elderly Care
7	s7	4500	Physiotherapy
8	s8	4000	Physical Therapy
9	s9	3000	Nursing
10	s10	2500	Elderly Care
11	s11	3500	Nursing
12	s12	3000	Elderly Care
13	s13	4000	Physiotherapy
14	s14	4500	Physical Therapy
15	s15	2000	Nursing
16	s16	5000	Elderly Care
17	s17	2500	Physiotherapy
18	s18	3000	Physical Therapy
19	s19	3500	Nursing

At the bottom of the results window, it says 'Total rows: 80 of 80' and 'Query complete 00:00:00.085'.

→ Find all appointments scheduled for a specific date.

```
SELECT * FROM "Homecare".Appointments  
WHERE "Date" = '2023-12-13';
```

Query Query History

```
1 SELECT * FROM "Homecare".Appointments
2 WHERE "Date" = '2023-12-13';
3
```

Data output Messages Notifications

≡

	Appointment_Id [PK] character varying	Service_Id character varying	Client_Id character varying	Date date	Status character varying
1	a9	s9	c9	2023-12-13	pending

→ List all clients who have pending appointments.

```
SELECT clients.*  
FROM "Homecare".Client clients  
JOIN "Homecare".Appointments appointments  
ON clients."Client_Id" = appointments."Client_Id"  
WHERE appointments."Status" = 'pending';
```

Query History

```
1 SELECT clients.*  
2 FROM "Homecare".Client clients  
3 JOIN "Homecare".Appointments appointments  
4 ON clients."Client_Id" = appointments."Client_Id"  
5 WHERE appointments."Status" = 'pending';  
6
```

Data output Messages Notifications

	Client_Id [PK] character varying	First_Name character varying	Last_Name character varying	Street character varying	City character varying	Pincode integer	Email character varying	Gender character varying
1	c1	Aarav	Kumar	Surya Nagar	Mumbai	400001	aarav.kumar@exa...	male
2	c2	Aarya	Sharma	Lotus Colony	Delhi	110001	aarya.sharma@exa...	female
3	c4	Amit	Patel	River View	Ahmedabad	380001	amit.patel@exa...	male
4	c5	Anaya	Singh	Golden Avenue	Chennai	600001	anaya.singh@exa...	female
5	c6	Arjun	Gupta	Heritage Road	Kolkata	700001	arjun.gupta@exa...	male
6	c7	Avni	Shah	Diamond Lane	Hyderabad	500001	avni.shah@exam...	female
7	c9	Esha	Rao	Jade Lane	Surat	395001	esha.rao@examp...	female
8	c11	Kabir	Deshpande	Topaz Lane	Lucknow	226001	kabir.deshpande...	male
9	c12	Kavya	Reddy	Amber Street	Nagpur	440001	kavya.reddy@exa...	female
10	c14	Mira	Chatterjee	Platinum Avenue	Vadodara	390001	mira.chatterjee@...	female
11	c16	Om	Dixit	Ruby Lane	Bhopal	462001	om.dixit@exampl...	male
12	c17	Pari	Sharma	Opal Street	Patna	800001	pari.sharma@exa...	female
13	c19	Riya	Chopra	Amethyst Road	Agra	282001	riya.chopra@exa...	female
14	c21	Saanvi	Khan	Emerald Lane	Varanasi	221001	saanvi.khan@exa...	female
15	c22	Shaurya	Nair	Silver Lane	Madurai	625001	shaurya.nair@ex...	male
16	c24	Tanvi	Prasad	Ruby Street	Allahabad	211001	tanvi.prasad@ex...	female
17	c26	Yash	Kumar	Sapphire Street	Srinagar	190001	yash.kumar@exa...	male
18	c27	Zara	Jain	Platinum Lane	Jodhpur	342001	zara.jain@examp...	female
19	c29	Aadi	Dubey	Coral Avenue	Firozabad	283203	aadi.dubey@exa...	male

Total rows: 37 of 37 | Query complete 00:00:00.061

→ Retrieve the total payment of a specific bill.

```
SELECT "Total_payment" FROM "Homecare".bill  
WHERE "Apppointment_id" = 'a5';
```

Query Query History

```
1 SELECT "Total_payment" FROM "Homecare".bill
2 WHERE "Apppointment_id" = 'a5';|
3
```

Data output Messages Notifications

≡+

	Total_payment	locked
1	4000	

→ Find the appointment with the highest total payment.

```
SELECT * FROM "Homecare".bill  
ORDER BY "Total_payment" DESC  
LIMIT 1;
```

Query Query History

```
1 SELECT * FROM "Homecare".bill  
2 ORDER BY "Total_payment" DESC  
3 LIMIT 1;  
4
```

Data output Messages Notifications

☰+ ↻ ↽ 🗑️ 🗑️ 🔍 🔍 ↴ ⚡

	Apppointment_id character varying	Total_payment integer	Bill_Date date	Payment_status character varying
1	a3	5000	2024-01-01	done

→ List all appointments with a "complete" status.

```
SELECT * FROM "Homecare".Appointments  
WHERE "Status" = 'complete';
```

Query Query History

```

1 SELECT * FROM "Homecare".Appointments
2 WHERE "Status" = 'complete';
3

```

Data output Messages Notifications

	Appointment_Id [PK] character varying	Service_Id character varying	Client_Id character varying	Date date	Status character varying
1	a3	s3	c3	2023-12-07	complete
2	a5	s5	c5	2023-12-09	complete
3	a8	s8	c8	2023-12-12	complete
4	a10	s10	c10	2023-12-14	complete
5	a13	s13	c13	2023-12-17	complete
6	a15	s15	c15	2023-12-19	complete
7	a18	s18	c18	2023-12-22	complete
8	a20	s20	c20	2023-12-24	complete
9	a23	s23	c23	2023-12-27	complete
10	a25	s25	c25	2023-12-29	complete
11	a28	s28	c28	2024-01-01	complete
12	a30	s30	c30	2024-01-03	complete
13	a33	s33	c33	2024-01-06	complete
14	a35	s35	c35	2024-01-08	complete
15	a38	s38	c38	2024-01-11	complete
16	a40	s40	c40	2024-01-13	complete
17	a45	s3	c45	2024-01-18	complete
18	a49	s5	c49	2024-01-22	complete
19	a53	s7	c53	2024-01-26	complete

Total rows: 25 of 25 Query complete 00:00:00.068

→ Find the caregiver with the most appointments.

```
SELECT caregivers."Caregiver_Id",COUNT(appointments."Appointment_Id") AS  
"Appointment_Count"  
FROM "Homecare".caregivers caregivers  
JOIN "Homecare".service service  
ON caregivers."Caregiver_Id" = service."Caregiver_Id"  
JOIN "Homecare".appointments appointments  
ON appointments."Service_Id"= service."Service_Id"  
GROUP BY caregivers."Caregiver_Id"  
ORDER BY "Appointment_Count" DESC  
LIMIT 10
```

The screenshot shows a database query editor interface. At the top, there are tabs for 'Query' and 'Query History'. Below the tabs, the SQL query is displayed in a code editor with line numbers from 1 to 10. Line 10 shows a progress indicator '>Loading...'. Below the code editor, there are tabs for 'Data output', 'Messages', and 'Notifications'. The 'Data output' tab is selected, displaying a table with two columns: 'Caregiver_Id' and 'Appointment_Count'. The data is as follows:

	Caregiver_Id	Appointment_Count
1	car3	4
2	car5	4
3	car12	3
4	car4	3
5	car7	3
6	car11	3
7	car17	3
8	car10	3
9	car18	3
10	car19	3

→ Find the clients who provided a rating below 3 in feedback.

```
SELECT clients.* , feedback."Rating"
FROM "Homecare".Client clients
JOIN "Homecare".Appointments appointments
ON clients."Client_Id" = appointments."Client_Id"
JOIN "Homecare".Feedback feedback
ON appointments."Appointment_Id" = feedback."Appoinment_id"
WHERE feedback."Rating" < 3;
```

Query Query History

```
1  SELECT clients.* , feedback."Rating"
2  FROM "Homecare".Client clients
3  JOIN "Homecare".Appointments appointments
4  ON clients."Client_Id" = appointments."Client_Id"
5  JOIN "Homecare".Feedback feedback
6  ON appointments."Appointment_Id" = feedback."Appoinment_id"
7  WHERE feedback."Rating" < 3;
8
```

Data output Messages Notifications

	Client_Id character varying	First_Name character varying	Last_Name character varying	Street character varying	City character varying	Pincode integer	Email character varying	Gender character varying	Rating integer
1	c15	Neha	Bose	Pearl Street	Kanpur	208001	neha.bose@exa...	female	2
2	c33	Ayaan	Chowdhury	Silver Avenue	Guntur	522001	ayaan.chowdhur...	male	2
3	c53	Ahaana	Sharma	Jade Street	Belagavi	590001	ahaana.sharma...	female	2
4	c69	Navya	Kumar	Topaz Avenue	Kakinada	533001	navya.kumar@ex...	female	2
5	c44	Saisha	Nair	Silver Street	Shimla	171001	saisha.nair@exa...	female	0

→ Get the total number of clients in the system.

SELECT COUNT(*) AS "Total_Clients" FROM "Homecare".Client;

Query Query History

```
1 SELECT COUNT(*) AS "Total_Clients" FROM "Homecare".Client;
2
```

Data output Messages Notifications

≡+

	Total_Clients
1	90

→ List caregivers who have not received any feedback.

```
SELECT caregivers.*  
FROM "Homecare".caregivers caregivers  
LEFT JOIN "Homecare".Feedback feedback  
ON caregivers."Caregiver_Id" = feedback."Appoinment_id"  
WHERE feedback."Appoinment_id" IS NULL;
```

Query Query History

```
1  SELECT caregivers.*  
2  FROM "Homecare".caregivers caregivers  
3  LEFT JOIN "Homecare".Feedback feedback  
4  ON caregivers."Caregiver_Id" = feedback."Appoinment_id"  
5  WHERE feedback."Appoinment_id" IS NULL;  
6
```

Data output Messages Notifications

	Caregiver_Id [PK] character varying	First_Name character varying	Last_Name character varying	Dob date	Gender character varying	Email character varying
1	car1	Rahul	Sharma	1980-05-12	male	rahul.sharma@ex...
2	car2	Priya	Gupta	1982-07-23	female	priya.gupta@exa...
3	car3	Amit	Verma	1981-12-15	male	amit.verma@exa...
4	car4	Sonia	Singh	1985-09-30	female	sonia.singh@exa...
5	car5	Vikram	Patel	1979-04-19	male	vikram.patel@ex...
6	car6	Anita	Rao	1983-03-25	female	anita.rao@exam...
7	car7	Raj	Malhotra	1980-11-09	male	raj.malhotra@ex...
8	car8	Meera	Kaur	1984-08-14	female	meera.kaur@exa...
9	car9	Rohan	Chopra	1982-06-27	male	rohan.chopra@e...
10	car10	Sneha	Deshpande	1986-02-08	female	sneha.deshpand...
11	car11	Sandeep	Bose	1981-01-21	male	sandeep.bose@e...
12	car12	Neha	Reddy	1983-10-17	female	neha.reddy@exa...
13	car13	Aryan	Joshi	1985-11-04	male	aryan.joshi@exa...
14	car14	Pooja	Shah	1982-09-09	female	pooja.shah@exa...
15	car15	Aakash	Gandhi	1980-12-31	male	aakash.gandhi@...
16	car16	Anjali	Srinivasan	1984-07-28	female	anjali.srinivasan...
17	car17	Alok	Iyer	1981-06-18	male	alok.iyer@examp...
18	car18	Smita	Chatterjee	1983-02-14	female	smita.chatterjee...
19	car19	Rajesh	Venkatesh	1985-04-02	male	rajesh.venkatesh...

Total rows: 90 of 90 Query complete 00:00:00.068

→ Find all bills with a "done" payment status.

```
SELECT * FROM "Homecare".bill  
WHERE "Payment_status" = 'done';
```

Query Query History

```
1  SELECT * FROM "Homecare".bill  
2  WHERE "Payment_status" = 'done';  
3  |  
4
```

Data output Messages Notifications

☰ 📁 ⏮ 🗑️ 📁 ⏴ ⏴

	Apppointment_id character varying	Total_payment integer	Bill_Date date	Payment_status character varying
1	a3	5000	2024-01-01	done
2	a8	4500	2024-01-03	done
3	a13	3500	2024-01-05	done
4	a18	2500	2024-01-07	done
5	a20	4500	2024-01-08	done
6	a25	3000	2024-01-10	done
7	a30	5000	2024-01-12	done
8	a33	2000	2024-01-13	done
9	a38	4500	2024-01-15	done
10	a40	4000	2024-01-16	done
11	a49	3500	2024-01-18	done
12	a57	2000	2024-01-20	done
13	a61	2500	2024-01-21	done
14	a69	4000	2024-01-23	done
15	a73	3000	2024-01-24	done

→ Calculate the total earnings of a specific caregiver.

```
SELECT SUM(Service."Charges") AS "Total_Earnings"  
FROM "Homecare".Service  
WHERE Service."Caregiver_Id" = 'car2';
```

Query Query History

```
1 SELECT SUM(Service."Charges") AS "Total_Earnings"
2 FROM "Homecare".Service
3 WHERE Service."Caregiver_Id" = 'car2';
4
```

>Loading...

Data output Messages Notifications

≡+ ↻ ✓ 🗑️ 📁 ⏴ ⏵

	Total_Earnings	🔒
1	10000	

→ Find the top 5 caregivers with the highest earnings.

```
SELECT caregivers."Caregiver_Id", SUM(Service."Charges") AS "Total_Earnings"
FROM "Homecare".caregivers caregivers
JOIN "Homecare".Service Service
ON caregivers."Caregiver_Id" = Service."Caregiver_Id"
GROUP BY caregivers."Caregiver_Id"
ORDER BY "Total_Earnings" DESC
LIMIT 5;
```

Query Query History

```
1 SELECT caregivers."Caregiver_Id", SUM(Service."Charges") AS "Total_Earnings"
2 FROM "Homecare".caregivers caregivers
3 JOIN "Homecare".Service Service
4 ON caregivers."Caregiver_Id" = Service."Caregiver_Id"
5 GROUP BY caregivers."Caregiver_Id"
6 ORDER BY "Total_Earnings" DESC
7 LIMIT 5;
8
```

Data output Messages Notifications

≡+ ↻ 📁 ↴ 🗑️ 📁 ↴ ↴

	Caregiver_Id [PK] character varying	Total_Earnings bigint
1	car14	12500
2	car16	11500
3	car5	10500
4	car10	10500
5	car17	10500

→ Identify clients who have appointments on multiple dates.

```
SELECT c."Client_Id", c."First_Name", c."Last_Name"
FROM "Homecare".Client c
JOIN "Homecare".Appointments a ON a."Client_Id" = c."Client_Id"
GROUP BY c."Client_Id", c."First_Name", c."Last_Name"
HAVING COUNT(DISTINCT a."Date") > 1;
```

Query Query History

```
1 SELECT c."Client_Id", c."First_Name", c."Last_Name"
2 FROM "Homecare".Client c
3 JOIN "Homecare".Appointments a ON a."Client_Id" = c."Client_Id"
4 GROUP BY c."Client_Id", c."First_Name", c."Last_Name"
5 HAVING COUNT(DISTINCT a."Date") > 1;
6
```

>Loading...

Data output Messages Notifications

≡ ⌂ ↻ 🗑 📁 ↴ ⚡

	Client_Id [PK] character varying	First_Name character varying	Last_Name character varying
1	c29	Aadi	Dubey
2	c45	Shlok	Venkataraman
3	c5	Anaya	Singh

→ List clients who haven't scheduled any appointments.

```
SELECT clients.*
FROM "Homecare".Client clients
LEFT JOIN "Homecare".Appointments appointments
ON clients."Client_Id" = appointments."Client_Id"
WHERE appointments."Client_Id" IS NULL;
```

Query Query History

```
1 SELECT clients.*
2 FROM "Homecare".Client clients
3 LEFT JOIN "Homecare".Appointments appointments
4 ON clients."Client_Id" = appointments."Client_Id"
5 WHERE appointments."Client_Id" IS NULL;
6
```

Data output Messages Notifications

	Client_Id [PK] character varying	First_Name character varying	Last_Name character varying	Street character varying	City character varying	Pincode integer	Email character varying	Gender character varying
1	c80	Shaan	Saxena	Topaz Avenue	Kamarhati	700058	shaan.saxena@e...	male
2	c81	Shlok	Kumar	Sapphire Street	Kamarhati	700058	shlok.kumar@ex...	male
3	c82	Vihaan	Sharma	Platinum Avenue	Kamarhati	700058	vihaan.sharma@...	male
4	c83	Vivaan	Malhotra	Diamond Lane	Kamarhati	700058	vivaan.malhotra...	male
5	c84	Yashvi	Deshpande	Coral Street	Kamarhati	700058	yashvi.deshpand...	female
6	c85	Yuvaan	Nair	Amethyst Avenue	Kamarhati	700058	yuvaan.nair@exa...	male
7	c86	Advait	Venkataraman	Jade Street	Kanchipuram	631501	advait.venkatara...	male
8	c87	Ahaan	Ghosh	Emerald Avenue	Kanchipuram	631501	ahaan.ghosh@ex...	male
9	c88	Arnav	Rao	Silver Street	Kanchipuram	631501	arnav.rao@exam...	male
10	c89	Arya	Saxena	Gold Avenue	Kanchipuram	631501	arya.saxena@ex...	female
11	c90	Atharv	Kumar	Ruby Street	Kanchipuram	631501	atharv.kumar@ex...	male

→Find the service types provided by a specific caregiver.

```
SELECT DISTINCT "Service_Type" FROM "Homecare".Service  
WHERE "Caregiver_Id" = 'car1';
```

```
Query    Query History

1  SELECT DISTINCT "Service_Type" FROM "Homecare".Service
2 WHERE "Caregiver_Id" = 'car1'; |
3
```

Data output	Messages	Notifications
	     	 
	Service_Type character varying 	
1	Nursing	
2	Physiotherapy	

→ Calculate the total earnings for each caregiver and display it in descending order.

```
SELECT Service."Caregiver_Id", SUM(Service."Charges") AS "Total_Earnings"
FROM "Homecare".appointments appointments
JOIN "Homecare".Service Service
ON appointments."Service_Id" = Service."Service_Id"
GROUP BY Service."Caregiver_Id"
ORDER BY "Total_Earnings" DESC;
```

Query Query History

```
1 SELECT Service."Caregiver_Id", SUM(Service."Charges") AS "Total_Earnings"
2 FROM "Homecare".appointments appointments
3 JOIN "Homecare".Service Service
4 ON appointments."Service_Id" = Service."Service_Id"
5 GROUP BY Service."Caregiver_Id"
6 ORDER BY "Total_Earnings" DESC;
7
```

Data output Messages Notifications

	Caregiver_Id character varying	Total_Earnings bigint
1	car16	15000
2	car5	14000
3	car14	13500
4	car7	13500
5	car13	12000
6	car4	12000
7	car8	12000
8	car20	12000
9	car11	10500
10	car19	10500
11	car1	10000
12	car3	10000
13	car12	9000
14	car2	9000
15	car9	9000
16	car18	9000
17	car17	7500
18	car10	7500
19	car15	6000

Total rows: 40 of 40 Query complete 00:00:00.061

→ Retrieve the last appointment date for each client.

Query Query History

```
1 SELECT "Client_Id", MAX("Date") AS "Last_Appointment_Date"
2 FROM "Homecare".Appointments
3 GROUP BY "Client_Id";
4
5 |
```

Data output Messages Notifications

≡+

	Client_Id character varying	Last_Appointment_Date date
1	c20	2023-12-24
2	c61	2024-02-03
3	c49	2024-01-22
4	c35	2024-01-08
5	c34	2024-01-07
6	c44	2024-01-17
7	c52	2024-01-25
8	c37	2024-01-10
9	c68	2024-02-10
10	c5	2024-01-15
11	c45	2024-01-18
12	c75	2024-02-17
13	c73	2024-02-15
14	c54	2024-01-27
15	c18	2023-12-22
16	c11	2023-12-15
17	c74	2024-02-16
18	c4	2023-12-08
19	c24	2023-12-28

Total rows: 79 of 79 Query complete 00:00:00.074

→List caregivers who have provided services more than one services.

```
SELECT "Caregiver_Id"  
FROM "Homecare".Service  
GROUP BY "Caregiver_Id"  
HAVING COUNT(DISTINCT "Service_Type") > 1;
```

→ Calculate the total revenue for the Homecare service.

```
SELECT SUM("Total_payment") AS "Total_Revenue" FROM "Homecare".bill;
```

Query Query History

```
1 SELECT SUM("Total_payment") AS "Total_Revenue" FROM "Homecare".bill;
2
```

Data output Messages Notifications

≡+ ↻ ↺ ↻ ↴ ↵

	Total_Revenue	🔒
1	88500	

→ Identify clients with the most appointments in the past month.

```
SELECT "Client_Id", COUNT(*) AS "Appointment_Count"
FROM "Homecare".Appointments
WHERE "Date" >= CURRENT_DATE - INTERVAL '1 month'
GROUP BY "Client_Id"
ORDER BY "Appointment_Count" DESC
LIMIT 1;
```

```
Query   Query History

1  SELECT "Client_Id", COUNT(*) AS "Appointment_Count"
2  FROM "Homecare".Appointments
3  WHERE "Date" >= CURRENT_DATE - INTERVAL '1 month'
4  GROUP BY "Client_Id"
5  ORDER BY "Appointment_Count" DESC
6  LIMIT 1;
7
```

→Find the appointments that have the same service and date.

```
SELECT *
FROM "Homecare".Appointments
WHERE ("Service_Id", "Date") IN (
    SELECT "Service_Id", "Date"
    FROM "Homecare".Appointments
    GROUP BY "Service_Id", "Date"
    HAVING COUNT(*) > 1
);
```

Query Query History

```
1  SELECT *
2  FROM "Homecare".Appointments
3  WHERE ("Service_Id", "Date") IN (
4      SELECT "Service_Id", "Date"
5      FROM "Homecare".Appointments
6      GROUP BY "Service_Id", "Date"
7      HAVING COUNT(*) > 1
8  );
9
```

Data output Messages Notifications

	Appointment_Id [PK] character varying	Service_Id character varying	Client_Id character varying	Date date	Status character varying
1	a38	s38	c38	2024-01-11	complete
2	a82	s38	c29	2024-01-11	pending

→List caregivers with a "female" gender who have appointments.

```
SELECT caregivers.*  
from "Homecare".caregivers caregivers  
join "Homecare".service ON service."Caregiver_Id" = caregivers."Caregiver_Id"  
WHERE service."Service_Id"  
in (SELECT service."Service_Id"  
      from "Homecare".Appointments)  
      and caregivers."Gender" = 'female';
```

```
1 SELECT caregivers.*  
2 from "Homecare".caregivers caregivers  
3 join "Homecare".service ON service."Caregiver_Id" = caregivers."Caregiver_Id"  
4 WHERE service."Service_Id"  
5 in (SELECT service."Service_Id"  
6      from "Homecare".Appointments)  
7      and caregivers."Gender" = 'female';  
8  
9
```

Data output Messages Notifications

The screenshot shows a database interface with a toolbar at the top and a table below it. The table has columns: Caregiver_Id [PK] character varying, First_Name character varying, Last_Name character varying, Dob date, Gender character varying, and Email character varying. The data consists of 40 rows, each containing a unique identifier (car1 to car40), a first name, a last name, a date of birth, a gender (all female), and an email address. The email addresses are truncated with ellipses. At the bottom of the table, it says 'Total rows: 40 of 40' and 'Query complete 00:00:00.073'.

	Caregiver_Id [PK] character varying	First_Name character varying	Last_Name character varying	Dob date	Gender character varying	Email character varying
1	car2	Priya	Gupta	1982-07-23	female	priya.gupta@exa...
2	car4	Sonia	Singh	1985-09-30	female	sonia.singh@exa...
3	car6	Anita	Rao	1983-03-25	female	anita.rao@exam...
4	car8	Meera	Kaur	1984-08-14	female	meera.kaur@exa...
5	car10	Sneha	Deshpande	1986-02-08	female	sneha.deshpand...
6	car12	Neha	Reddy	1983-10-17	female	neha.reddy@exa...
7	car14	Pooja	Shah	1982-09-09	female	pooja.shah@exa...
8	car16	Anjali	Srinivasan	1984-07-28	female	anjali.srinivasan...
9	car18	Smita	Chatterjee	1983-02-14	female	smita.chatterjee...
10	car20	Kavita	Rao	1980-03-09	female	kavita.rao@exam...
11	car21	Amita	Saxena	1982-08-07	female	amita.saxena@e...
12	car23	Nisha	Nair	1984-11-28	female	nisha.nair@exam...
13	car25	Kiran	Rastogi	1983-06-29	female	kiran.rastogi@ex...
14	car26	Vidya	Sethi	1985-01-19	female	vidya.sethi@exa...
15	car29	Sunita	Malhotra	1984-10-08	female	sunita.malhotra...
16	car31	Ananya	Bose	1983-02-07	female	ananya.bose@ex...
17	car33	Kajal	Joshi	1981-08-09	female	kajal.joshi@exa...
18	car35	Riya	Gandhi	1980-06-22	female	riya.gandhi@exa...
19	car37	Sarita	Iyer	1981-10-13	female	sarita.iyer@exam...

→ List clients who have pending appointments and their contact details.

```
SELECT clients.* , appointments."Status"
FROM "Homecare".Client clients
JOIN "Homecare".Appointments appointments
ON clients."Client_Id" = appointments."Client_Id"
WHERE appointments."Status" = 'pending';
```

Query Query History

```
1 SELECT clients.* , appointments."Status"
2 FROM "Homecare".Client clients
3 JOIN "Homecare".Appointments appointments
4 ON clients."Client_Id" = appointments."Client_Id"
5 WHERE appointments."Status" = 'pending';
6
7
8 |
```

Data output Messages Notifications

	Client_Id character varying	First_Name character varying	Last_Name character varying	Street character varying	City character varying	Pincode integer	Email character varying	Gender character varying	Status character varying
1	c1	Aarav	Kumar	Surya Nagar	Mumbai	400001	aarav.kumar@example.com	male	pending
2	c2	Aarya	Sharma	Lotus Colony	Delhi	110001	aarya.sharma@example.com	female	pending
3	c4	Amit	Patel	River View	Ahmedabad	380001	amit.patel@example.com	male	pending
4	c5	Anaya	Singh	Golden Avenue	Chennai	600001	anaya.singh@example.com	female	pending
5	c6	Arjun	Gupta	Heritage Road	Kolkata	700001	arjun.gupta@example.com	male	pending
6	c7	Avni	Shah	Diamond Lane	Hyderabad	500001	avni.shah@example.com	female	pending
7	c9	Esha	Rao	Jade Lane	Surat	395001	esha.rao@example.com	female	pending
8	c11	Kabir	Deshpande	Topaz Lane	Lucknow	226001	kabir.deshpande@example.c...	male	pending
9	c12	Kavya	Reddy	Amber Street	Nagpur	440001	kavya.reddy@example.com	female	pending
10	c14	Mira	Chatterjee	Platinum Avenue	Vadodara	390001	mira.chatterjee@example.com	female	pending
11	c16	Om	Dixit	Ruby Lane	Bhopal	462001	om.dixit@example.com	male	pending
12	c17	Pari	Sharma	Opal Street	Patna	800001	pari.sharma@example.com	female	pending
13	c19	Riya	Chopra	Amethyst Road	Agra	282001	riya.chopra@example.com	female	pending
14	c21	Saanvi	Khan	Emerald Lane	Varanasi	221001	saanvi.khan@example.com	female	pending
15	c22	Shaurya	Nair	Silver Lane	Madurai	625001	shaurya.nair@example.com	male	pending
16	c24	Tanvi	Prasad	Ruby Street	Allahabad	211001	tanvi.prasad@example.com	female	pending
17	c26	Yash	Kumar	Sapphire Street	Srinagar	190001	yash.kumar@example.com	male	pending
18	c27	Zara	Jain	Platinum Lane	Jodhpur	342001	zara.jain@example.com	female	pending
19	c29	Aadi	Dubey	Coral Avenue	Firozabad	283203	aadi.dubey@example.com	male	pending

Total rows: 37 of 37 Query complete 00:00:00.070 Successfully run. Total query runtime: 70 msec. 37 rows affected.

→ Calculate the total number of completed appointments for each caregiver.

```
SELECT caregivers."Caregiver_Id", COUNT(*) AS "Completed_Appointments"
FROM "Homecare".caregivers caregivers join
"Homecare".service ON service."Caregiver_Id" = caregivers."Caregiver_Id"
join "Homecare".Appointments appointments on Appointments."Service_Id" =
service."Service_Id"
WHERE appointments."Status" = 'complete'
GROUP BY caregivers."Caregiver_Id";
```

Query Query History

```
1 SELECT caregivers."Caregiver_Id", COUNT(*) AS "Completed_Appointments"
2 FROM "Homecare".caregivers caregivers join
3 "Homecare".service ON service."Caregiver_Id" = caregivers."Caregiver_Id"
4 join "Homecare".Appointments appointments on Appointments."Service_Id" = service."Service_Id"
5 WHERE appointments."Status" = 'complete'
6 GROUP BY caregivers."Caregiver_Id";
7
```

Data output Messages Notifications

	Caregiver_Id [PK] character varying	Completed_Appointments bigint
1	car8	1
2	car5	2
3	car17	1
4	car7	1
5	car3	2
6	car25	1
7	car19	1
8	car15	2
9	car9	1
10	car38	1
11	car18	1
12	car35	1
13	car10	1
14	car33	1
15	car40	1
16	car20	1
17	car23	1
18	car28	1
19	car13	2

Total rows: 21 of 21 Query complete 00:00:00.043

→ Identify the most popular service type based on the number of appointments.

```
SELECT "Service_Type", COUNT(*) AS "Appointment_Count"
FROM "Homecare".Service
GROUP BY "Service_Type"
ORDER BY "Appointment_Count" DESC
LIMIT 1;
```

The screenshot shows a database query interface with the following components:

- Query Tab:** Contains the SQL code for the query.
- Data output Tab:** Contains the results of the query.
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste, Find, Refresh, Undo, Redo).
- Table:** Displays the results of the query with two columns: Service_Type and Appointment_Count.

	Service_Type	Appointment_Count
1	Nursing	26

→ List caregivers who have not been assigned any appointments.

```
SELECT caregivers.*  
FROM "Homecare".caregivers caregivers  
LEFT JOIN "Homecare".Appointments appointments  
ON caregivers."Caregiver_Id" = appointments."Service_Id"  
WHERE appointments."Service_Id" IS NULL;
```

Query Query History

```
1 SELECT caregivers.*  
2 FROM "Homecare".caregivers caregivers  
3 LEFT JOIN "Homecare".Appointments appointments  
4 ON caregivers."Caregiver_Id" = appointments."Service_Id"  
5 WHERE appointments."Service_Id" IS NULL;  
6 |
```

Data output Messages Notifications

	Caregiver_Id [PK] character varying	First_Name character varying	Last_Name character varying	Dob date	Gender character varying	Email character varying
1	car1	Rahul	Sharma	1980-05-12	male	rahul.sharma@ex...
2	car2	Priya	Gupta	1982-07-23	female	priya.gupta@exa...
3	car3	Amit	Verma	1981-12-15	male	amit.verma@exa...
4	car4	Sonia	Singh	1985-09-30	female	sonia.singh@exa...
5	car5	Vikram	Patel	1979-04-19	male	vikram.patel@ex...
6	car6	Anita	Rao	1983-03-25	female	anita.rao@exam...
7	car7	Raj	Malhotra	1980-11-09	male	raj.malhotra@ex...
8	car8	Meera	Kaur	1984-08-14	female	meera.kaur@exa...
9	car9	Rohan	Chopra	1982-06-27	male	rohan.chopra@e...
10	car10	Sneha	Deshpande	1986-02-08	female	sneha.deshpand...
11	car11	Sandeep	Bose	1981-01-21	male	sandeep.bose@e...
12	car12	Neha	Reddy	1983-10-17	female	neha.reddy@exa...
13	car13	Aryan	Joshi	1985-11-04	male	aryan.joshi@exa...
14	car14	Pooja	Shah	1982-09-09	female	pooja.shah@exa...
15	car15	Aakash	Gandhi	1980-12-31	male	aakash.gandhi@...
16	car16	Anjali	Srinivasan	1984-07-28	female	anjali.srinivasan...
17	car17	Alok	Iyer	1981-06-18	male	alok.iyer@examp...
18	car18	Smita	Chatterjee	1983-02-14	female	smita.chatterjee...
19	car19	Rajesh	Venkatesh	1985-04-02	male	rajesh.venkatesh...

Total rows: 90 of 90 Query complete 00:00:00.063

→ Find the appointments scheduled for clients with a specific pin code.

```
SELECT appointments.*  
FROM "Homecare".Appointments appointments  
JOIN "Homecare".Client clients  
ON appointments."Client_Id" = clients."Client_Id"  
WHERE clients."Pincode" = 394190;
```

The screenshot shows a database query interface with the following details:

- Query Tab:** The tab is selected, showing the SQL query:

```
1 SELECT appointments.*  
2 FROM "Homecare".Appointments appointments  
3 JOIN "Homecare".Client clients  
4 ON appointments."Client_Id" = clients."Client_Id"  
5 WHERE clients."Pincode" = 394190;  
6
```

A tooltip "Loading..." is visible near the WHERE clause.
- Data output Tab:** This tab is active, displaying the schema of the result set:

Appointment_Id	Service_Id	Client_Id	Date	Status
[PK] character varying	character varying	character varying	date	character varying
- Messages Tab:** Shows no messages.
- Notifications Tab:** Shows no notifications.

→ Calculate the total payments for "done" bills in the last month.

```
SELECT SUM("Total_payment") AS "Total_Payments"  
FROM "Homecare".bill  
WHERE "Payment_status" = 'done'  
AND "Bill_Date" >= CURRENT_DATE - INTERVAL '1 month';
```

Query Query History

```
1 SELECT SUM("Total_payment") AS "Total_Payments"
2 FROM "Homecare".bill
3 WHERE "Payment_status" = 'done'
4 AND "Bill_Date" >= CURRENT_DATE - INTERVAL '1 month';
5
```

 Loading...

Data output Messages Notifications

	Total_Payments	bigint
1	53500	

→List clients who provided feedback with a rating of 5.

```
SELECT clients.*  
FROM "Homecare".Client clients  
JOIN "Homecare".Appointments appointments  
ON clients."Client_Id" = appointments."Client_Id"  
JOIN "Homecare".Feedback feedback  
ON appointments."Appointment_Id" = feedback."Appoinment_id"  
WHERE feedback."Rating" = 5;
```

Query Query History

```
1 SELECT clients.*  
2 FROM "Homecare".Client clients  
3 JOIN "Homecare".Appointments appointments  
4 ON clients."Client_Id" = appointments."Client_Id"  
5 JOIN "Homecare".Feedback feedback  
6 ON appointments."Appointment_Id" = feedback."Appoinment_id"  
7 WHERE feedback."Rating" = 5;  
8
```

Data output Messages Notifications

	Client_Id [PK] character varying	First_Name character varying	Last_Name character varying	Street character varying	City character varying	Pincode integer	Email character varying	Gender character varying
1	c3	Aditi	Joshi	Rose Street	Bangalore	560001	aditi.joshi@exam...	female
2	c8	Dev	Verma	Silver Street	Pune	411001	dev.verma@exa...	male
3	c20	Rohan	Srinivasan	Jasper Avenue	Kochi	682001	rohan.srinivasan...	male
4	c30	Ahaan	Bhattacharya	Amethyst Lane	Bhubaneswar	751001	ahaan.bhattacha...	male
5	c40	Navya	Malhotra	Coral Street	Nashik	422001	navya.malhotra@...	female
6	c51	Aarna	Kulkarni	Coral Street	Aurangabad	431001	aarna.kulkarni@e...	female
7	c77	Sanskriti	Shah	Silver Street	Kamarhati	700058	sanskriti.shah@e...	female

→ Identify appointments that have a "complete" status but no feedback.

```
SELECT appointments.*  
FROM "Homecare".Appointments appointments  
LEFT JOIN "Homecare".Feedback feedback  
ON appointments."Appointment_Id" = feedback."Appoinment_id"  
WHERE appointments."Status" = 'complete' AND feedback."Appoinment_id" IS NULL;
```

Query Query History

```
1  SELECT appointments.*  
2  FROM "Homecare".Appointments appointments  
3  LEFT JOIN "Homecare".Feedback feedback  
4  ON appointments."Appointment_Id" = feedback."Appoinment_id"  
5  WHERE appointments."Status" = 'complete' AND feedback."Appoinment_id" IS NULL;  
6  |  
7  | Loading...
```

Data output Messages Notifications

	Appointment_Id [PK] character varying	Service_Id character varying	Client_Id character varying	Date date	Status character varying
1	a61	s11	c61	2024-02-03	complete

→ Find caregivers with complete appointments who have been rated poorly.

```
SELECT caregivers.*  
FROM "Homecare".caregivers caregivers  
JOIN "Homecare".service service  
ON caregivers."Caregiver_Id" = service."Caregiver_Id"  
JOIN "Homecare".appointments appointments  
ON appointments."Service_Id"= service."Service_Id"  
JOIN "Homecare".feedback feedback  
ON feedback."Appointment_id" = appointments."Appointment_Id"  
WHERE appointments."Status" = 'complete' AND feedback."Rating" < 3;
```

Query Query History

```
1  SELECT caregivers.*  
2  FROM "Homecare".caregivers caregivers  
3  JOIN "Homecare".service service  
4  ON caregivers."Caregiver_Id" = service."Caregiver_Id"  
5  JOIN "Homecare".appointments appointments  
6  ON appointments."Service_Id"= service."Service_Id"  
7  JOIN "Homecare".feedback feedback  
8  ON feedback."Appointment_id" = appointments."Appointment_Id"  
9  WHERE appointments."Status" = 'complete' AND feedback."Rating" < 3;  
10
```

Loading...

Data output Messages Notifications

	Caregiver_Id [PK] character varying	First_Name character varying	Last_Name character varying	Dob date	Gender character varying	Email character varying
1	car15	Aakash	Gandhi	1980-12-31	male	aakash.gandhi@...
2	car33	Kajal	Joshi	1981-08-09	female	kajal.joshi@exa...
3	car7	Raj	Malhotra	1980-11-09	male	raj.malhotra@ex...
4	car15	Aakash	Gandhi	1980-12-31	male	aakash.gandhi@...

→ Calculate the average rating for each caregiver.

```
SELECT caregivers."Caregiver_Id", AVG(feedback."Rating")AS "Average_Rating"
FROM "Homecare".caregivers caregivers
JOIN "Homecare".service service
ON caregivers."Caregiver_Id" = service."Caregiver_Id"
JOIN "Homecare".appointments appointments
ON appointments."Service_Id"= service."Service_Id"
JOIN "Homecare".feedback feedback
ON feedback."Appointment_id" = appointments."Appointment_Id"
GROUP BY caregivers."Caregiver_Id";
```

Query Query History

```
1
2 SELECT caregivers."Caregiver_Id", AVG(feedback."Rating")AS "Average_Rating"
3 FROM "Homecare".caregivers caregivers
4 JOIN "Homecare".service service
5 ON caregivers."Caregiver_Id" = service."Caregiver_Id"
6 JOIN "Homecare".appointments appointments
7 ON appointments."Service_Id"= service."Service_Id"
8 JOIN "Homecare".feedback feedback
9 ON feedback."Appointment_id" = appointments."Appointment_Id"
10 GROUP BY caregivers."Caregiver_Id";|
```

Data output Messages Notifications

	Caregiver_Id [PK] character varying	Average_Rating numeric
1	car5	3.50000000000000
2	car7	2.00000000000000
3	car18	3.00000000000000
4	car35	3.00000000000000
5	car10	3.00000000000000
6	car28	4.00000000000000
7	car17	4.00000000000000
8	car3	3.00000000000000
9	car19	5.00000000000000
10	car40	5.00000000000000
11	car23	4.00000000000000
12	car13	3.50000000000000
13	car30	5.00000000000000
14	car8	5.00000000000000
15	car9	4.00000000000000
16	car38	4.00000000000000
17	car6	5.00000000000000
18	car33	2.00000000000000
19	car20	5.00000000000000

Total rows: 21 of 21 Query complete 00:00:00.066

→ Identify the services provided on a specific date.

```
SELECT s."Service_Type"
FROM "Homecare".service s
JOIN "Homecare".appointments a ON s."Service_Id" = a."Service_Id"
WHERE a."Date" = '2024-02-07';
```

Query Query History

```
1 SELECT s."Service_Type"
2 FROM "Homecare".service s
3 JOIN "Homecare".appointments a ON s."Service_Id" = a."Service_Id"
4 WHERE a."Date" = '2024-02-07';
5 Loading...
```

Data output Messages Notifications

Service_Type
character varying

	Service_Type
1	Physiotherapy

→ Calculate the total earnings for each service type.

```
SELECT s."Service_Type", SUM(b."Total_payment") AS Total_Earnings
FROM "Homecare".service s
JOIN "Homecare".appointments a ON a."Service_Id" = s."Service_Id"
JOIN "Homecare".bill b ON b."Apppoiment_id" = a."Appointment_Id"
GROUP BY s."Service_Type";
```

Query Query History

```
1  SELECT s."Service_Type", SUM(b."Total_payment") AS Total_Earnings
2  FROM "Homecare".service s
3  JOIN "Homecare".appointments a ON a."Service_Id" = s."Service_Id"
4  JOIN "Homecare".bill b ON b."Apppoiment_id" = a."Appointment_Id"
5  GROUP BY s."Service_Type";
6  Loading...
```

Data output Messages Notifications

	Service_Type	totalLearnings
1	Elderly Care	24500
2	Physical Therapy	15000
3	Physiotherapy	22000
4	Nursing	27000

→List clients who have provided feedback and their ratings.

```
SELECT c."Client_Id", c."First_Name", c."Last_Name", f."Rating"  
FROM "Homecare".client c  
JOIN "Homecare".appointments a ON c."Client_Id" = a."Client_Id"  
JOIN "Homecare".feedback f ON a."Appointment_Id" = f."Appoinment_id";
```

Query Query History

```
1 SELECT c."Client_Id", c."First_Name", c."Last_Name", f."Rating"  
2 FROM "Homecare".client c  
3 JOIN "Homecare".appointments a ON c."Client_Id" = a."Client_Id"  
4 JOIN "Homecare".feedback f ON a."Appointment_Id" = f."Appoinment_id";  
5
```

Data output Messages Notifications

	Client_Id character varying	First_Name character varying	Last_Name character varying	Rating integer
1	c3	Aditi	Joshi	5
2	c5	Anaya	Singh	4
3	c8	Dew	Verma	5
4	c10	Ishaan	Malhotra	3
5	c13	Leela	Gandhi	4
6	c15	Neha	Bose	2
7	c18	Pranav	Iyer	3
8	c20	Rohan	Srinivasan	5
9	c23	Shreya	Venkatesh	4
10	c25	Vivaan	Saxena	3
11	c28	Zoya	Venkataraman	4
12	c30	Ahaan	Bhattacharya	5
13	c33	Ayaan	Chowdhury	2
14	c35	Ira	Gupta	3
15	c38	Manan	Chopra	4
16	c40	Navya	Malhotra	5
17	c45	Shlok	Venkataraman	4
18	c49	Amita	Sethi	3
19	c53	Ahaana	Sharma	2

Total rows: 26 of 26 Query complete 00:00:00.058

→Identify the most profitable service based on the total payment.

```
SELECT s."Service_Type", SUM(b."Total_payment") AS Total_Earnings
FROM "Homecare".service s
JOIN "Homecare".appointments a ON a."Service_Id" = s."Service_Id"
JOIN "Homecare".bill b ON b."Apppoiment_id" = a."Appointment_Id"
GROUP BY s."Service_Type"
ORDER BY Total_Earnings DESC
LIMIT 1;
```

Query Query History

```
1 SELECT s."Service_Type", SUM(b."Total_payment") AS Total_Earnings
2 FROM "Homecare".service s
3 JOIN "Homecare".appointments a ON a."Service_Id" = s."Service_Id"
4 JOIN "Homecare".bill b ON b."Apppoiment_id" = a."Appointment_Id"
5 GROUP BY s."Service_Type"
6 ORDER BY Total_Earnings DESC
7 LIMIT 1;
8
Loading...
```

Data output Messages Notifications

	Service_Type	total_earnings
1	Nursing	27000

→Find the average total payment for appointments for each city.

```
SELECT c."City", AVG(b."Total_payment") AS Average_Payment
FROM "Homecare".client c
JOIN "Homecare".appointments a ON a."Client_Id" = c."Client_Id"
JOIN "Homecare".bill b ON b."Apppoiment_id" = a."Appointment_Id"
GROUP BY c."City";
```

Query Query History

```
1  SELECT c."City", AVG(b."Total_payment") AS Average_Payment
2  FROM "Homecare".client c
3  JOIN "Homecare".appointments a ON a."Client_Id" = c."Client_Id"
4  JOIN "Homecare".bill b ON b."Apppoiment_id" = a."Appointment_Id"
5  GROUP BY c."City";
```

Data output Messages Notifications

	City character varying	average_payment numeric
1	Jaipur	3000.000000000000
2	Kochi	4500.000000000000
3	Guna	2500.000000000000
4	Muzaffarnagar	4500.000000000000
5	Durgapur	2000.000000000000
6	Ludhiana	2500.000000000000
7	Bhubaneswar	5000.000000000000
8	Kamarhati	3250.000000000000
9	Thanjavur	3000.000000000000
10	Belagavi	5000.000000000000
11	Kanpur	2000.000000000000
12	Chennai	4000.000000000000
13	Nashik	4000.000000000000
14	Jamnagar	2500.000000000000
15	Udaipur	3500.000000000000
16	Indore	3500.000000000000
17	Bangalore	5000.000000000000
18	Amritsar	3000.000000000000
19	Pune	4500.000000000000

Total rows: 24 of 24 Query complete 00:00:00.058

Triggers and functions

→ **Stored Procedure 1: Calculate a caregiver's total earnings.**

```
CREATE OR REPLACE FUNCTION CalculateTotalEarnings(CaregiverId character varying,  
OUT TotalEarnings integer)  
AS $$  
BEGIN  
    SELECT SUM(s."Charges") INTO TotalEarnings  
    FROM "Homecare".service s  
    WHERE s."Caregiver_Id" = CaregiverId;  
END;  
$$ LANGUAGE plpgsql;
```

```
select * from calculatetotalearnings('car3');
```

```
CREATE OR REPLACE FUNCTION CalculateTotalEarnings(CaregiverId character varying, OUT TotalEarnings integer)  
AS $$  
BEGIN  
    SELECT SUM(s."Charges") INTO TotalEarnings  
    FROM "Homecare".service s  
    WHERE s."Caregiver_Id" = CaregiverId;  
END;  
$$ LANGUAGE plpgsql;
```



```
select * from calculatetotalearnings('car3');
```

output	Messages	Notifications

totalearnings	9500
integer	

→**Stored Procedure 2: Calculate the average rating for each caregiver.**

```
CREATE OR REPLACE FUNCTION CalculateAverageRating(CaregiverId character varying,  
OUT AverageRating numeric)  
AS $$  
BEGIN  
SELECT AVG(f."Rating") INTO AverageRating  
FROM "Homecare".feedback f  
JOIN "Homecare".appointments a ON f."Appointment_id" = a."Appointment_Id"  
JOIN "Homecare".service s ON s."Service_Id" = a."Service_Id"  
WHERE s."Caregiver_Id" = CaregiverId;  
END;  
$$ LANGUAGE plpgsql;
```

```
select * from calculateaveragerating('car3');
```

The screenshot shows a PostgreSQL query editor interface. The top section displays the SQL code for creating a function named 'CalculateAverageRating'. The function takes a character varying parameter 'CaregiverId' and returns a numeric value 'AverageRating' through an OUT parameter. The body of the function performs a query to calculate the average rating for a specific caregiver by joining three tables: 'feedback', 'appointments', and 'service'. The bottom section shows the result of executing the function with the argument 'car3', which returns a single row with an average rating of 3.000000000000.

	averagerating
1	3.000000000000

→ **Stored Procedure 3: Generate a monthly report of earnings for caregivers.**

```
CREATE OR REPLACE FUNCTION GenerateMonthlyReport(Month integer, Year integer)
RETURNS TABLE("Caregiver_Id" character varying, "TotalEarnings" integer) AS $$  
BEGIN  
    RETURN QUERY  
        SELECT s."Caregiver_Id", SUM(s."Charges")::integer AS "TotalEarnings"  
        FROM "Homecare".service s  
        JOIN "Homecare".appointments a ON s."Service_Id" = a."Service_Id"  
        WHERE EXTRACT(MONTH FROM a."Date") = Month AND EXTRACT(YEAR FROM a."Date")  
        = Year  
        GROUP BY s."Caregiver_Id";  
END;  
$$ LANGUAGE plpgsql;
```

```
select * from generatemonthlyreport(12, 2023);
```

Query Query History

```
1 CREATE OR REPLACE FUNCTION GenerateMonthlyReport(Month integer, Year integer)
2 RETURNS TABLE("Caregiver_Id" character varying, "TotalEarnings" integer) AS $$  
3 BEGIN  
4     RETURN QUERY  
5     SELECT s."Caregiver_Id", SUM(s."Charges")::integer AS "TotalEarnings"  
6     FROM "Homecare".service s  
7     JOIN "Homecare".appointments a ON s."Service_Id" = a."Service_Id"  
8     WHERE EXTRACT(MONTH FROM a."Date") = Month AND EXTRACT(YEAR FROM a."Date") = Year  
9     GROUP BY s."Caregiver_Id";  
10 END;  
11 $$ LANGUAGE plpgsql;  
12  
13  
14  
15 select * from generatemonthlyreport(12, 2023);  
16  
17 |  
  ::Loading...
```

Data output Messages Notifications

File Edit View Insert Tools Window Help

	Caregiver_Id character varying	TotalEarnings integer
1	car1	5000
2	car10	2500
3	car11	3500
4	car12	3000
5	car13	4000
6	car14	4500
7	car15	2000
8	car16	5000
9	car17	2500
10	car18	3000
11	car19	3500

→**Stored Procedure 4: Find and list clients with no scheduled appointments.**

```
CREATE OR REPLACE FUNCTION FindClientsWithNoAppointments()
RETURNS TABLE("Client_Id" character varying, "First_Name" character varying, "Last_Name"
character varying) AS $$

BEGIN
    RETURN QUERY
    SELECT c."Client_Id", c."First_Name", c."Last_Name"
    FROM "Homecare".client c
    WHERE c."Client_Id" NOT IN (
        SELECT DISTINCT a."Client_Id"
        FROM "Homecare".appointments a
    );
END;
$$ LANGUAGE plpgsql;
```

select * from FindClientsWithNoAppointments();

The screenshot shows a PostgreSQL query editor interface. At the top, there are tabs for 'Query' (which is selected) and 'Query History'. Below the tabs is a code editor area containing the SQL code for creating the function and executing it. The code is numbered from 1 to 19. Lines 17 and 18 show the execution of the function. At the bottom, there is a 'Data output' tab, a toolbar with various icons, and a results table displaying the names of 11 clients who have no scheduled appointments. The results table has columns: Client_Id, First_Name, and Last_Name. The data is as follows:

	Client_Id	First_Name	Last_Name
1	c80	Shaan	Saxena
2	c81	Shlok	Kumar
3	c82	Vihaan	Sharma
4	c83	Vivaan	Malhotra
5	c84	Yashvi	Deshpande
6	c85	Yuvaan	Nair
7	c86	Adyait	Venkataraman
8	c87	Ahaan	Ghosh
9	c88	Arnav	Rao
10	c89	Arya	Saxena
11	c90	Atharv	Kumar

Total rows: 11 of 11 Query complete 00:00:05.148

→**Trigger 1: Automatically add a bill for completed appointments with a payment status of "pending."**

```
CREATE OR REPLACE FUNCTION AddPendingBillForCompletedAppointments()
RETURNS TRIGGER AS $$

BEGIN
    IF NEW."Status" = 'complete' THEN
        INSERT INTO "Homecare".bill ("Apppointment_id", "Total_payment", "Bill_Date",
"Payment_status")
        VALUES (NEW."Appointment_Id", 0, CURRENT_DATE, 'pending');
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER AddPendingBillForCompletedAppointmentsTrigger
AFTER INSERT ON "Homecare".appointments
FOR EACH ROW
EXECUTE FUNCTION AddPendingBillForCompletedAppointments();
```

insert INTO "Homecare".appointments VALUES ('a88', 's6', 'c47', '2024-01-20', 'complete');

select * from "Homecare".bill b where b."Apppoiment_id" = 'a88';



	Apppoiment_id	Total_payment	Bill_Date	Payment_status
1	a88	0	2023-11-08	pending

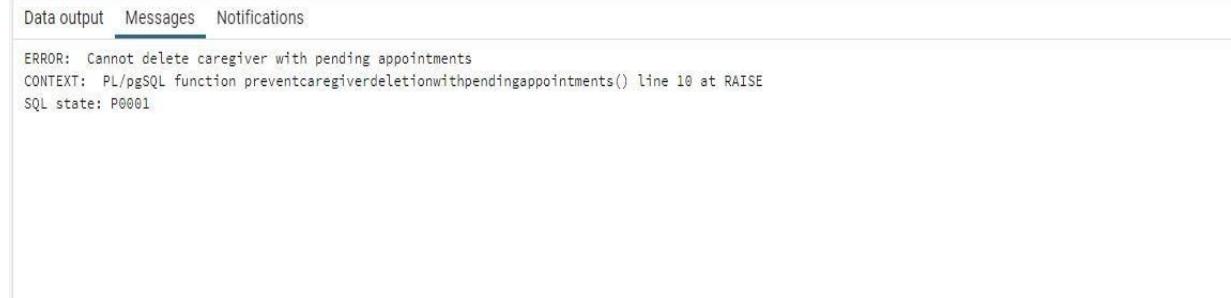
→ **Trigger 2: Prevent the deletion of a caregiver record if they have pending appointments.**

```
CREATE OR REPLACE FUNCTION PreventCaregiverDeletionWithPendingAppointments()
RETURNS TRIGGER AS $$

BEGIN
    IF EXISTS (
        SELECT 1
        FROM "Homecare".appointments a
        JOIN "Homecare".service s ON a."Service_Id" = s."Service_Id"
        JOIN "Homecare".caregivers c ON c."Caregiver_Id" = s."Caregiver_Id"
        WHERE c."Caregiver_Id" = OLD."Caregiver_Id" AND a."Status" = 'pending'
    ) THEN
        RAISE EXCEPTION 'Cannot delete caregiver with pending appointments';
    END IF;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER PreventCaregiverDeletionTrigger
BEFORE DELETE ON "Homecare".caregivers
FOR EACH ROW
EXECUTE FUNCTION PreventCaregiverDeletionWithPendingAppointments();
```

```
DELETE FROM "Homecare".caregivers
WHERE "Caregiver_Id" = 'car1';
```



The screenshot shows a PostgreSQL terminal window. At the top, there are three tabs: 'Data output' (disabled), 'Messages' (selected), and 'Notifications'. Below the tabs, an error message is displayed:

```
ERROR: Cannot delete caregiver with pending appointments
CONTEXT: PL/pgSQL function preventcaregiverdeletionwithpendingappointments() line 10 at RAISE
SQL state: P0001
```

