

Final Project Submission

for

Home Care Worker Providing Agency Database Management System

Version 1.6 Approved

Prepared by

Ashutosh Anand 202318035

Rishi Pawar 202318037

18th November 2023

Software Requirements Specification

Contents

1. Descriptions	4
1.1 Purpose:.....	4
1.2 Audience and Reading Suggestions:.....	4
1.3 Product Scope:	5
1.4 Home Care Industry:.....	5
1.5 Product's Perspective:	6
1.6 Problems Faced by Client and Caregiver Before the Product:	6
1.7 Problems Being Solved by the Product:	6
1.8 Product Functions:	7
1.9 References:.....	8
2. Background Reading:.....	8
2.1.1 Description of Reading:	8
2.1.2 References.....	11
2.2 Interview:	11
2.2.1 Interview Plan:.....	11
2.2.2 Requirements Gathered:.....	13
2.3 Questionnaire/s	15
2.3.1 Caregiver Questionnaire:	15
2.3.2 Client Questionnaire:	20
2.3.3 Requirements Gathered from the Questionnaire:	25
2.4 Observations:	27
2.4.1 Summary of observation/s:	27
2.4.2 Combined requirements gathered from the observation:.....	28
2.5 Fact Finding Chart	29
3. List of Requirements:	30
4. User categories and privileges:	31
4.1 User Cases:.....	31
4.2 Pre-Post Conditions of User Cases:	32
5. List of Assumptions:	33
6. Business Constraints:.....	34
7. Noun & Verb Analysis:.....	36
8. ERD:.....	46
9. Normalized Schema:.....	49
10. DDL Scripts:.....	49
11. Populating the Tables:.....	54
12. SQL Queries:.....	60

Revision History

Name	Date	Reason For Changes	Version
Home Care Worker System	23-08-23	Initial Version	0.1
Home Care Worker System	10-09-2023	Added Introduction and Overall Description for the project	0.5
Home Care Worker System	17-09-2023	Added Interview, Questionnaire Requirements	1.0
Home Care Worker System	21-09-2023	Added Assumptions and Constraints	1.5
Home Care Worker System	18-11-2023	Revised SRS	1.6

1. Descriptions

1.1 Purpose:

The core objectives of this document are multifaceted, primarily aiming to:

- Articulate the sequential development of the foundational requirements for the Home Care Worker Providing Agency Database Management System.
- Present the specifications for the Database Management System in a manner that delineates the transformation of requirements from their embryonic, vague phase to a clear, well-defined state.
- Illustrate our comprehensive consideration and rigorous analysis of the needs, demonstrating how dependencies were identified, problems were resolved, and the depth of our understanding of the requirements.
- Consequently, this document is not merely a compilation of requirements but also serves as a narrative detailing the derivation, interrelation, and refinement of these requirements from the initial inputs received.

1.2 Audience and Reading Suggestions:

- Entities committed to providing personalized care services to individuals within their homes, employing a cadre of caregivers and health practitioners to render diverse, specialized assistance.
- This demographic includes individuals receiving home care services, such as the elderly, the differently-abled, and those in medical recuperation. Their families, who are integral to the caregiving decisions, are also encompassed within this audience.
- This group consists of medical specialists and dedicated caregivers, including doctors, therapists, and nurses, who synergize with home care organizations to deliver bespoke, expert care solutions to clients.

1.3 Product Scope:

- The Home Care Worker Providing Agency Database Management System is designed to envelop a plethora of functionalities, enabling the meticulous management of the agency's diverse operational facets. Below is an enhanced overview of the system's comprehensive scope:
 - Preserve extensive client profiles encompassing both personal and medical details.
 - Facilitate optimal client-caregiver matches, according to preferences and needs, while enabling smooth communication for scheduling and feedback.
 - Sustain exhaustive worker profiles detailing qualifications and availability and orchestrate their schedules to ensure seamless client service.
 - Monitor the progress and status of worker training and certifications.
 - Formulate and modify individualized care plans, adapting to the fluctuating needs of clients.
 - Consolidate billing details, process invoices, and oversee payment information, while generating insightful reports for compliance and analytical purposes.
 - Allocate shifts, provide instantaneous updates, and maintain a user-friendly mobile interface to enhance user convenience and accessibility.
 - Uphold stringent data security measures and conduct quality assurance to gather constructive feedback for continual service improvement.
 - Adapt to organizational growth, manage operational expenses effectively, and leverage data-driven insights to facilitate informed strategic decisions.

1.4 Home Care Industry:

The Home Care Industry is a rapidly growing sector within healthcare, providing essential services to individuals who require assistance with their daily living activities. This industry primarily focuses on delivering care services to the elderly, disabled, or chronically ill individuals who prefer to stay in their homes rather than move to nursing homes or healthcare facilities. Services offered by the home care industry include personal care, medical care, meal preparation, housekeeping, and companionship.

1.5 Product's Perspective:

In this context, the product is a sophisticated database system designed to optimize the operation of home care service agencies. It aims to streamline service allocation, improve communication, enhance user satisfaction, and ensure the security of sensitive information. The product seeks to cater to the unique needs and preferences of caregivers, clients, and administrative staff, focusing on user-centric design, advanced feature integration, and effective feedback mechanisms.

1.6 Problems Faced by Client and Caregiver Before the Product:

Before the introduction of this product, clients and caregivers in the home care industry were grappling with several challenges:

- Clients and caregivers often struggled with mismatched service allocations, where caregivers were assigned to clients whose needs did not align with their skills or preferences.
- Inadequate communication channels led to misunderstandings and inefficiencies, impacting the quality of care provided and the overall satisfaction of both clients and caregivers.
- Caregivers often felt unsupported and undertrained, struggling to adapt to new technologies and lacking the necessary skills to meet the diverse needs of their clients.
- The absence of robust security measures raised concerns about the confidentiality and integrity of sensitive information, undermining trust between clients and caregivers.
- The lack of a user-friendly, mobile-responsive platform made it challenging for users to manage tasks, receive updates, and access essential information on the go.

1.7 Problems Being Solved by the Product:

This product is designed to address the challenges by offering the following solutions:

- The system ensures accurate and efficient service allocation, matching caregivers with clients based on their skills, experience, and service preferences, thereby enhancing service quality and user satisfaction.

- By providing multiple communication options and streamlined reporting mechanisms, the product facilitates clear and effective communication between clients, caregivers, and administrative staff, reducing misunderstandings and improving operational efficiency.
- The product assesses and improves the training and support provided to users, ensuring that caregivers are well-equipped to handle diverse client needs and can effectively utilize the system.
- The integration of advanced security features protects user information and builds trust by ensuring the confidentiality and security of sensitive data.
- The development of a user-centric, intuitive interface and a responsive mobile app enhances user experience and accessibility, allowing users to manage tasks and receive updates efficiently through their preferred platforms.
- The incorporation of effective feedback mechanisms and analytical tools enables continuous improvement and prioritization of feature development based on user needs and preferences, ensuring the system remains relevant and user-friendly.

1.8 Product Functions:

- Administrators enrich the system by populating it with various caregiving specialties pertinent to the home care industry, ensuring a comprehensive representation of all available services and care types.
- The system meticulously links these specialties to the respective caregivers, forming a robust foundation for the caregiver database, allowing for precise and efficient service allocations.
- Integration of Chemist and Lab Information:
 - The system integrates detailed information from local chemists and labs into their designated databases, providing a consolidated view of all available medical resources and facilities in the vicinity.
 - Home care workers register within the system, detailing their professional qualifications, experience, availability, and preferred tasks, forming a comprehensive caregiver profile.
 - Clients seeking caregiving services are facilitated to register and specify their care requirements, preferences, and schedules, ensuring a customized and client-centric approach to service allocation.

- The system employs advanced matching algorithms to pair clients with the most suitable caregivers, considering factors like specialty, availability, client needs, and preferences, ensuring optimal satisfaction and service quality.
- Caregivers are granted access to enriched client profiles, complete with detailed care needs, schedules, and specific instructions, fostering a well-informed and efficient caregiving process.
- Administrative users are endowed with extensive management and oversight capabilities, enabling them to create and manage user accounts, configure system settings, and access advanced analytics to monitor and optimize caregiver performance and agency operations.
- The system is built with a user-centric approach, focusing on creating a seamless and intuitive user experience for both caregivers and clients, while also providing administrators with the tools necessary to manage and optimize the system effectively.

1.9 References:

- IEEE 830-1998
- <https://medium.com/@f20180617/business-model-of-urban-clap-company-c342485c1b49>
- <https://www.document-logistix.com/hospitality/hotels/restaurants/events/document-management-for-hotels.html>
- <https://demigos.com/blog-post/how-to-develop-an-it-solution-that-saves-management-effort-for-a-caregiving-company/>
- <https://www.osplabs.com/insights/how-to-develop-home-healthcare-application-from-scratch/>
- <https://www.g2.com/categories/home-health-care>

2. Background Reading:

2.1.1 Description of Reading:

Books:

1. Database Systems: Design, Implementation, and Management" by Carlos Coronel, Steven Morris, and Peter Rob:

It provides a comprehensive understanding of designing, implementing, and managing databases. While not tailored specifically to home care worker agencies, the book covers essential concepts such as data modeling, relational databases, SQL, security, integrity, and performance tuning. It can offer valuable insights for building a robust and efficient database management system to handle client records, staff information, scheduling, billing, communication, and more within a home care worker providing agency context.

2. Home Care How To" by Brendan John:

- The book emphasizes the importance of a well-thought-out plan, laying the foundation for the agency's mission, vision, and long-term objectives.
- Insights on essential legal prerequisites, documentation, and regulations crucial for running a compliant in-home care service.
- Detailed steps on setting up an efficient workflow, ensuring smooth day-to-day running of the agency.
- Guidelines on recruiting skilled and compassionate professionals, maintaining the quality and reputation of the service.
- Strategies to develop personalized care plans, addressing the unique needs and preferences of each senior.
- Tips on promoting the agency, reaching out to potential clients, and building a strong brand presence in the market.
- Expertise on budgeting, billing, financial record-keeping, and ensuring the agency's profitability.

3. The Complete Eldercare Planner: Where to Start, Which Questions to Ask, and How to Find Help" by Joy Loverde:

- Offers insights into various care possibilities, helping families make informed decisions about residential care, home care, and specialized services.
- Addresses critical topics like power of attorney, medical directives, patient rights, and navigating medical treatments.
- Provides strategies for managing stress and ensuring mental wellness for both caregivers and the elderly.
- The book's insights can enhance an agency's ability to comprehensively address client needs, adhere to legal standards, and coordinate effective medical care.

4. "Business Intelligence Guidebook: From Data Integration to Analytics" by Rick Sherman:

- Data Integration and Warehousing**: Delivers insights on consolidating various data sources, ensuring seamless data storage, retrieval, and analysis in the context of client profiles and staff performance.
- Emphasizes deriving actionable insights from data, allowing for effective staff scheduling, performance tracking, and quality assurance measures.
- Highlights the importance of maintaining data integrity, security, and consistent management practices for an efficient database system.

-Equips readers with knowledge on the latest tools and practices, driving innovation in data-driven strategies to improve agency performance and client care.

Articles:

1. Design and Development of a Web-Based Home Healthcare Information System –
 - Lifestyle-related diseases like stroke, heart disease, and cancer are closely related to daily lifestyle choices.
 - Security attributes like integrity, availability, and confidentiality must be safeguarded.
 - Users can access their data from anywhere via the Internet and receive specialist feedback.
 - The article concludes with the implementation of a prototype WBHMS, emphasizing secure communication protocols.
 - Users can access their data from anywhere via the Internet and receive specialist feedback. The system uses server computers with open-source software, ensuring data security and backup.
2. Home care clients: a research protocol -
 - Utilizes an encompassing socio-ecological model to scrutinize the system of home care specifically designed for senior citizens.
 - Centers attention on two principal data streams, namely Client/Service Data and Constellation Data.
 - The Constellation Data Stream delves into the experiences encountered by clients, caregivers, healthcare professionals, and workers.
 - Employs qualitative research methods to delve into the myriad factors impacting service pathways, striving to understand the intricate relationships and influences.
3. The Future of Home Health Care:
 - The role of agencies certified by Medicare is crucial, serving as the linchpin in achieving this vision by delivering superior and efficient care within home and community settings.
 - Given the preference among recipients for home care due to its cost-efficient and high-quality services, it is instrumental in fostering patient-centric and economical healthcare solutions.
 - The role of agencies certified by Medicare is crucial, serving as the linchpin in achieving this vision by delivering superior and efficient care within home and community settings.
4. Optimizing Weekend Schedules in Home Health Care:
 - The Essential Care on Weekends (ECoW) strategy is central to refining scheduling practices in home health care, specifically targeting limitations in service capacity prevalent during weekends.
 - ECoW's deployment underscores the viability of refining scheduling protocols to amplify the capacity of home-based care, guaranteeing that at-risk clients procure the requisite

support to stay within their communities. This methodology is in harmony with the objectives of a Home Care Worker Providing Agency DB Management System, promoting the judicious distribution of resources and enhancing the caliber of care dispensed.

5. A qualitative study of home care client:

- The study underscores the beneficial influences of the DIVERT-CARE intervention on participants' abilities to manage their own care, educational attainment, and emotional wellness.
- Such a system should be designed to streamline communication, disseminate education efficiently, and cultivate enriching interactions between patients and providers, thereby refining the experiences of home-based care recipients.

6. Planning the Episode: Home Care Admission Nurse Decision-Making:

- The study illuminates the intricate decision-making pathways that nurses navigate to establish visit schedules, a process shaped by the conditions of the patients and the accessibility of pertinent information.
- The insights gained underscore the critical need for robust information transfer and decision assistance within a Home Care Worker Providing Agency DB Management System, enabling the formulation of precise and suitable scheduling of visits.

2.1.2 References

- "Database Systems: Design, Implementation, and Management" by Carlos Coronel, Steven Morris, and Peter Rob
- "Home Care How To: The Guide to Starting Your Senior in Home Care Business" by Brendan John:
- "The Complete Eldercare Planner: Where to Start, Which Questions to Ask, and How to Find Help" by Joy Loverde:
- [Design and development of a Web-based hospital information system](#)
- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7291732/>
- <https://journals.sagepub.com/doi/10.1177/1084822316666368>
- <https://journals.sagepub.com/doi/10.1177/10848223231183091>
- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8103627/>
- <https://journals.sagepub.com/doi/full/10.1177/1084822321990775>

2.2 Interview:

2.2.1 Interview Plan:

System: Home Care Worker Providing Agency Database Management

Participants: Ashutosh Anand(G21), Kunal Anand (Owner), Anurag Shukla (Co-Owner)

Date: 17/9/2023

Time: 12:30**Duration:** 30 minutes**Place:** DS Lab**Purpose of Interview:**

- To delve into the operational dynamics of an agency providing home care services and gain a profound understanding of its functionalities.
- To acquaint oneself with the prevailing working environments and the routine processes undertaken daily within the agency.
- To grasp the complexities and challenges encountered by the proprietor of the agency in navigating the provision of home care services.
- To comprehend the apprehensions and considerations of the service providers within the agency's framework.
- To procure in-depth knowledge and insights into the intricate workings of the agency, discerning the nuances of its operational structure.
- To pinpoint potential realms requiring refinement and augmentation, aiming to elevate the standards of service provision.
- To facilitate enhancements in the overall proficiency and efficacy of the services rendered in home care, contributing to the betterment of care quality and delivery.

Interview Agenda:

- Insight into the agency's origins and mission.
- Outline of principal services provided.
- Overview of the agency's scale and reach.
- In-depth exploration of daily functionalities and operational procedures.
- Examination of workflow strategies and standard practices.
- Highlighting of distinctive or exemplary operational methods.
- Reflection on the hurdles, ambitions, and goals from the standpoint of the agency proprietor.
- Deliberation on pivotal strategic intentions and plans.
- Review of the apprehensions and feedback from the agency's service personnel.
- Investigation into the staff's viewpoints on the operational aspects of the agency.
- Recognition of potential areas necessitating refinement.
- Proposals for advancements in operational efficacy and productivity.

Question Asked in interview carried out by Role Playing:

- Have there been remarkable cases or conflicts where a structured complaint resolution process would have been beneficial?
- What approaches are you contemplating to set up a proficient resolution system?
- Can you explain how newly recruited employees are delegated specific tasks within the agency?
- Have there been instances or concerns regarding new staff members being allocated challenging tasks?
- What methodologies could be implemented to guarantee optimal task assignment for newcomers?
- Could you shed light on any disparities in skills among your workforce?
- How does the inequitable spread of skills impact the functioning of the agency?
- Are there any strategies or programs in place to balance the skill distribution within the team?
- Can you identify the services that are perceived to pose significant occupational risks?
- What factors contribute to the lack of insurance provisions for employees engaged in high-risk services?
- Can you outline the procedures your agency currently employs for maintaining records?
- Have you experienced any challenges or restrictions due to the absence of electronic record-keeping?
- Are there certain categories of records that you think should be digitized as a priority?
- Could you delve into how customers usually schedule services with your agency?
- Have you noted any inefficiencies or obstacles in the process of manual reservations?
- Have manual bookings ever resulted in overlooked opportunities or inaccuracies?
- Could you elaborate on how feedback from customers regarding provided services is currently obtained?

2.2.2 Requirements Gathered:

1. Intelligent Caregiver-Client Matching System:

Develop advanced algorithms focusing on the intelligent matching of caregivers and clients, considering their individual skills, needs, and specializations, to ensure optimal assignment and enhanced service quality.

2. Efficient Scheduling and Conflict Resolution:

Design a robust system capable of generating efficient schedules while minimizing overlaps and conflicts, with automated reminders and notifications for shifts and updates, ensuring seamless operations and enhanced productivity.

3. Enhanced Data Security and Privacy:

Implement stringent measures to safeguard sensitive caregiver and client information, ensuring data security and privacy, and compliance with relevant regulations.

4. Comprehensive Reporting and Analytics:

Incorporate detailed and intuitive reporting features to monitor caregiver performance, client satisfaction, and agency management, allowing for informed decision-making and continuous improvement.

5. Real-Time Updates and Communications:

Provide instant notifications and real-time updates to all stakeholders, enabling clients to monitor care provision and facilitating direct communication between clients and caregivers.

6. User-Friendly Mobile App:

Develop a user-friendly mobile app for caregivers and clients, allowing easy access to schedules, communications, and updates, and enabling clients to request services and provide feedback conveniently.

7. Automated Service Booking and Digital Calendar Integration:

Implement an automated service booking feature that allows clients to schedule services online efficiently, integrated with a digital calendar system to manage bookings and allocate service providers accurately.

8. Formal Customer Feedback and Grievance Redressal Mechanism:

Establish formal mechanisms for collecting and analyzing customer feedback and resolving grievances from both customers and service providers, contributing to service improvement and enhanced customer satisfaction.

9. Expertise Distribution and Development:

Implement strategies for training and development to balance the distribution of expertise among employees, focusing on areas where expertise is lacking, to improve service quality and employee skillsets.

10. Insurance and Safety Provisions for High-Risk Roles:

Review and establish policies for providing insurance and safety provisions to employees in high-risk service roles, identifying high-risk roles, and determining the level of coverage to be provided.

11. Digitization and Prioritization of Records:

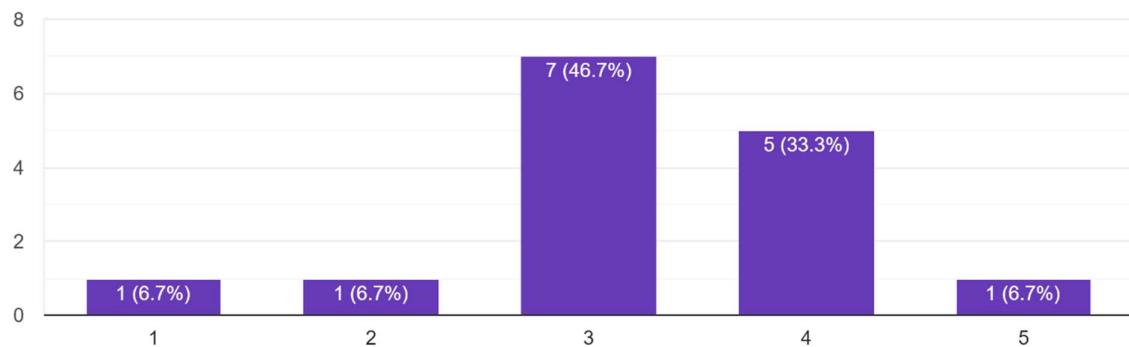
Develop a system for digitizing agency records efficiently, specifying the types of records to be digitized first based on their operational impact, to ensure streamlined and efficient record-keeping.

These requirements will serve as a comprehensive framework for developing the Home Care Worker Database Management System.

2.3 Questionnaire/s**2.3.1 Caregiver Questionnaire:**

How would you rate your current workload on a scale of 1 to 5?

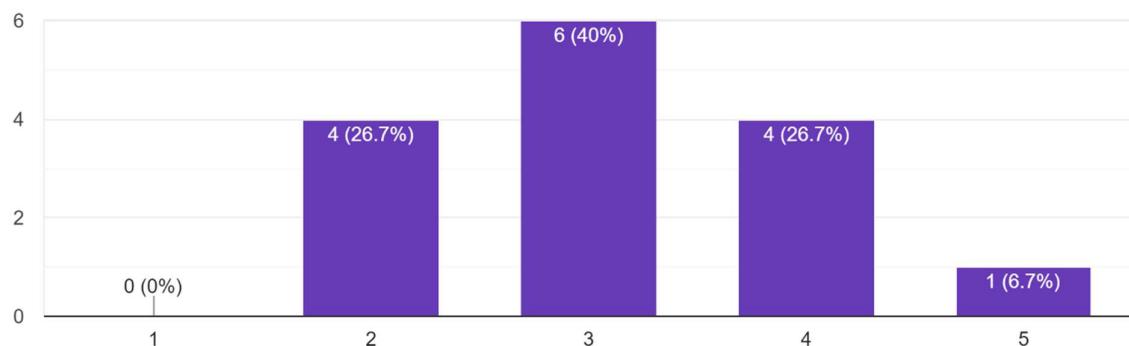
15 responses



Most caregivers have rated their current workload as average to high, indicating a balanced to a slightly heavy workload. This may imply a need to reevaluate assignment processes to ensure workload equity.

How good is the client matching process in the system?

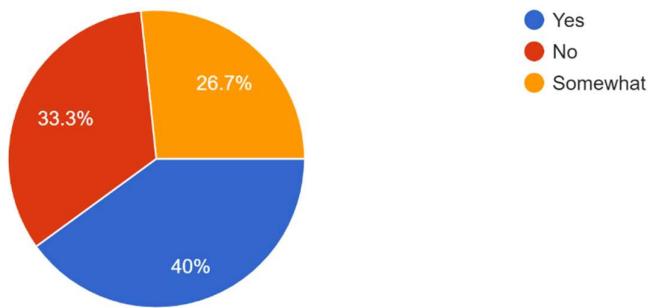
15 responses



Responses varied, but a significant number found the client matching process satisfactory. This indicates that while the current matching process is functional, there is room for improvement to enhance caregiver satisfaction.

Do you feel the agency provides sufficient training and opportunities for professional development to keep your skills updated and relevant?

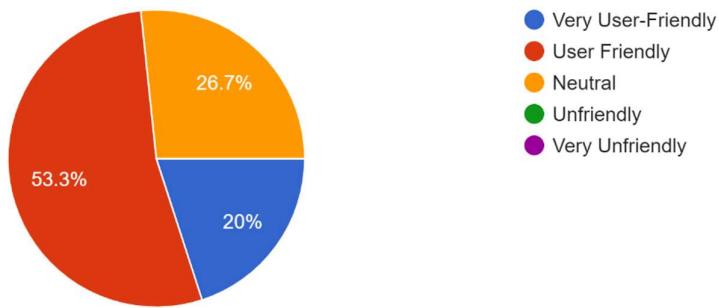
15 responses



Mixed responses were recorded, with some caregivers finding the training and development opportunities sufficient, while others did not. This suggests a need for more consistent and comprehensive training and development programs.

How user-friendly do you find the technology and software tools provided by the agency for managing work tasks?

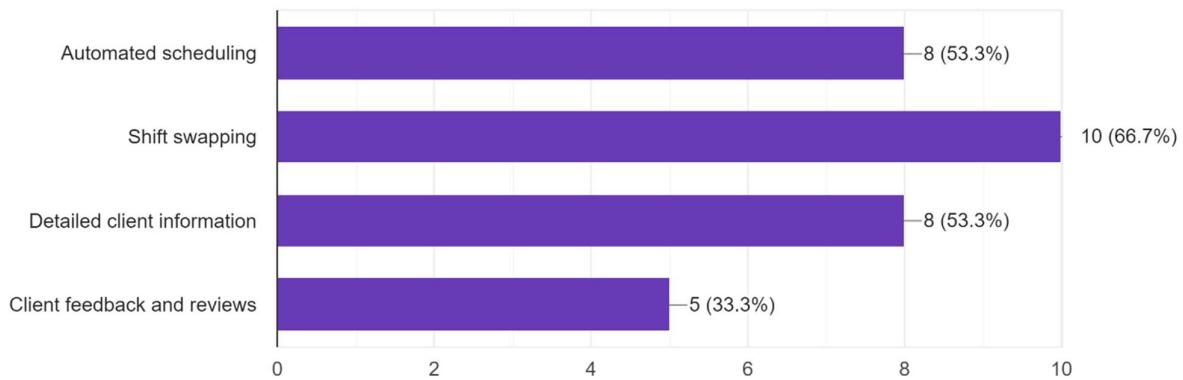
15 responses



The majority found the provided tools to be user-friendly, but a few indicated neutrality. This implies that while the technology is generally accessible, further refinements can be made to improve user experience and accessibility.

What feature of the system will be more efficient for you as a caregiver? (Select all that apply)

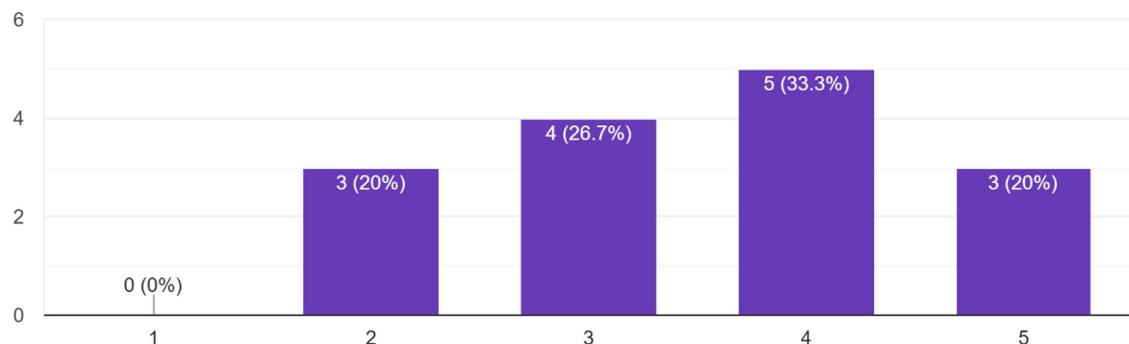
15 responses



Automated scheduling and shift swapping were frequently selected as preferable features, highlighting a desire for more efficient and flexible scheduling options. Detailed client information and client feedback were also valued, indicating the importance of comprehensive and relevant client data.

How satisfied are you with the current matching with clients based on your skills and preferences?

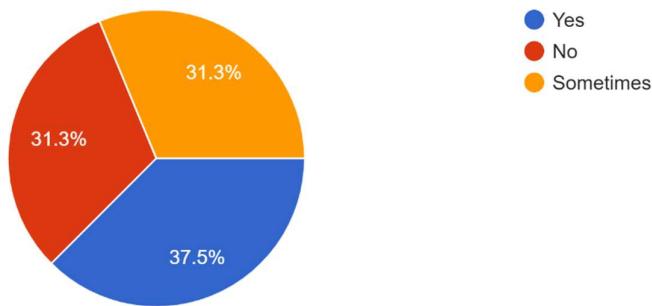
15 responses



Most caregivers expressed moderate satisfaction with how they are matched with clients based on their skills and preferences. This reveals a potential area for enhancement in the client matching algorithm to improve caregiver satisfaction.

Are you comfortable in travelling far from your location?

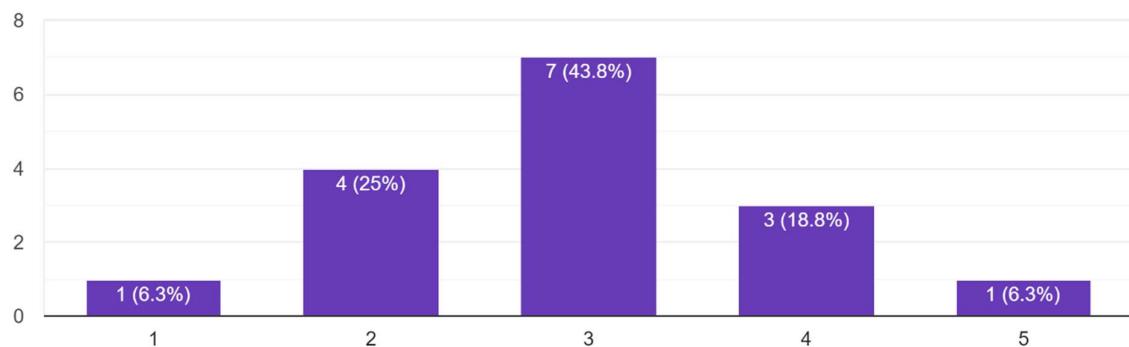
16 responses



The responses are varied, with some caregivers comfortable with traveling, some sometimes comfortable, and others not comfortable. This suggests a need for a more personalized approach to assigning clients, considering individual preferences regarding travel.

How do you feel that the security measures in place to protect caregiver are robust and reliable.

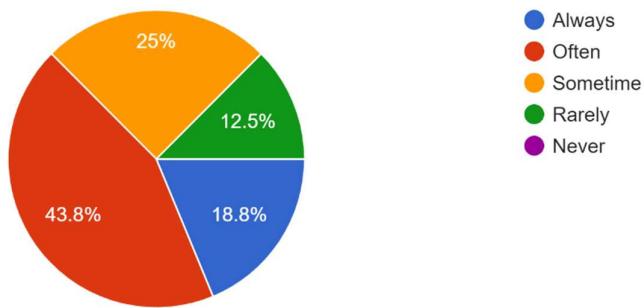
16 responses



Generally, caregivers seem to have a moderate level of trust in the security measures in place, with room for improvement to reinforce caregiver confidence in the system's security protocols.

Do you feel the agency accommodates your shift preferences and provides sufficient flexibility in scheduling?

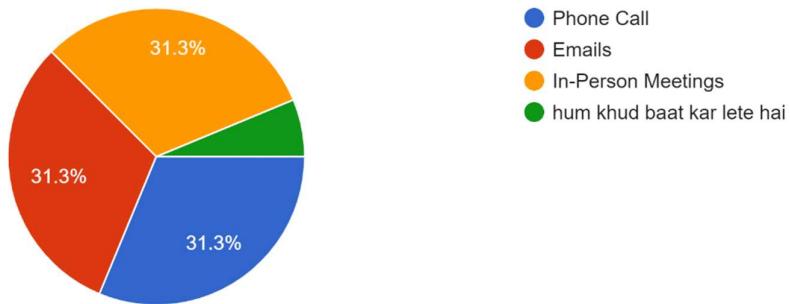
16 responses



A significant number of caregivers felt that their shift preferences are often accommodated, but some rarely find flexibility in scheduling. This indicates a need to enhance scheduling flexibility to accommodate individual preferences and needs.

How do you usually convey concerns or issues related to your work or clients to the agency?

16 responses

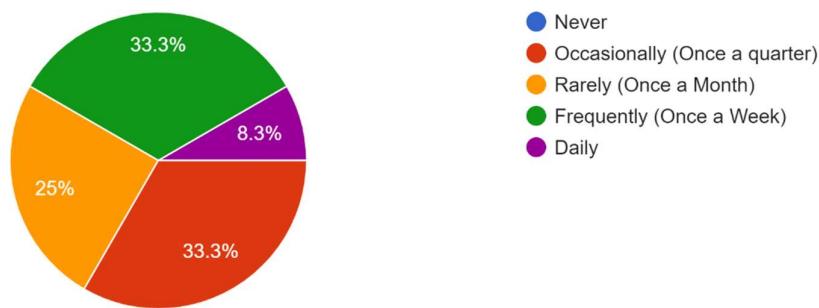


Caregivers use various methods to convey their concerns, with in-person meetings and emails being the most common. This suggests that multiple channels of communication should be maintained, but also streamlined, to ensure effective communication of concerns and issues.

2.3.2 Client Questionnaire:

How often do you interact with a home care worker scheduling system ?

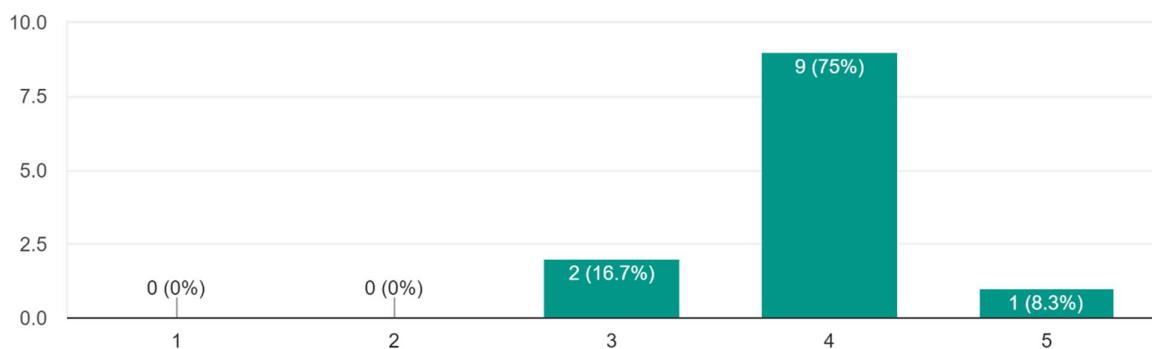
12 responses



The frequency of interaction with the home care worker scheduling system varies among clients. Some interact daily or weekly, while others use it rarely or occasionally. This implies a diverse range of needs and preferences among clients in terms of scheduling frequency.

How satisfied are you with the current booking process?

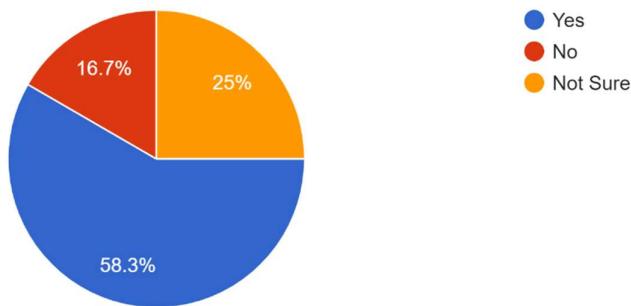
12 responses



Most clients seem to be satisfied with the current booking process, but improvements could enhance user experience. This suggests a need to continually refine and optimize the booking process based on user feedback.

Do you think the current system effectively matches caregivers with clients based on their skills and needs?

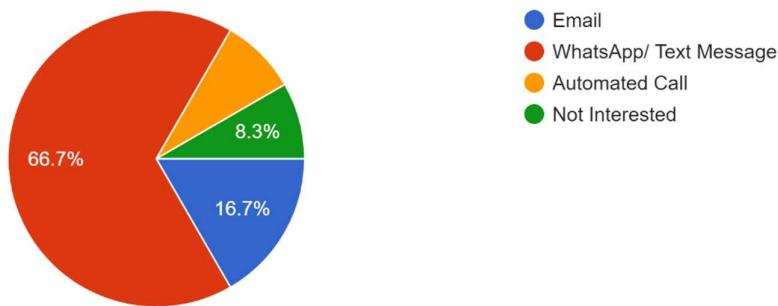
12 responses



Responses are mixed, with some clients uncertain or unsatisfied with how effectively the system matches caregivers with clients. This highlights a potential area for enhancement in the matching algorithm to better align with client needs and preferences.

Which platform you prefer to provide feedback and reviews about caregivers?

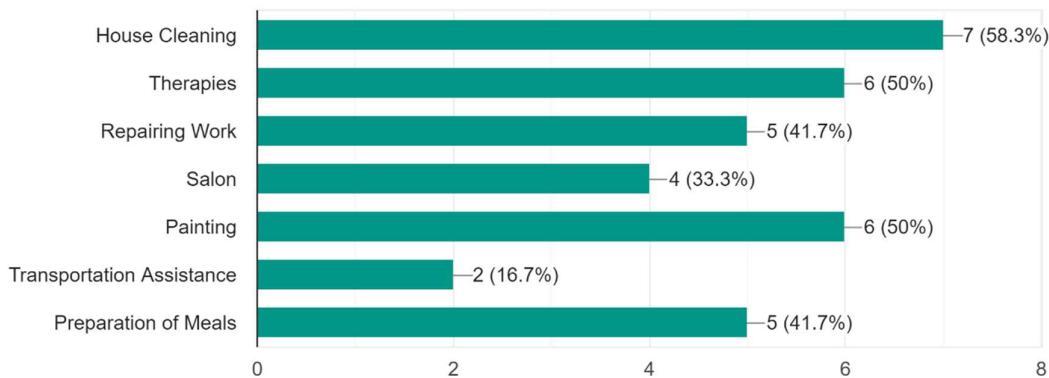
12 responses



Email and WhatsApp/Text Message are the preferred platforms for providing feedback and reviews about caregivers. This indicates that digital and instant communication platforms are favoured for feedback provision.

Which of these services have you booked recently?

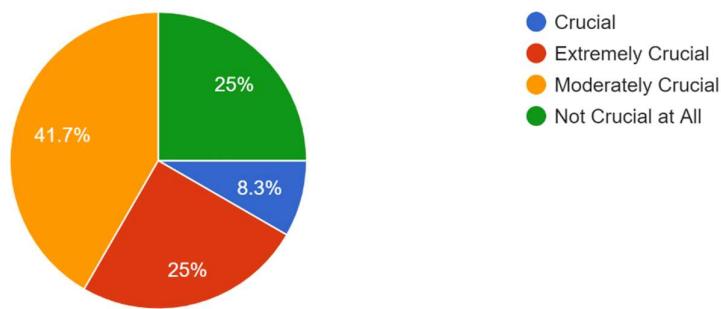
12 responses



Clients have booked a variety of services, ranging from house cleaning and therapies to painting and meal preparation. This suggests diverse service needs among clients, necessitating a versatile service offering.

How crucial is it for you to receive services from the same caregiver each time?

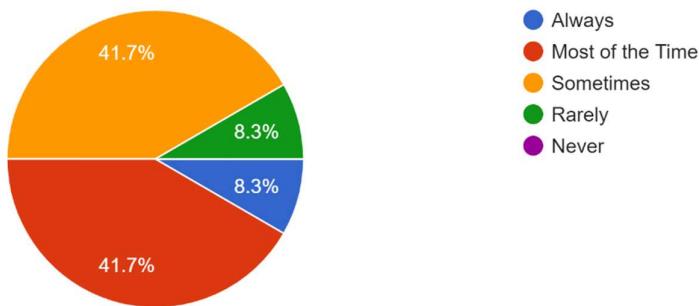
12 responses



The importance of receiving services from the same caregiver each time varies, with some clients finding it extremely crucial while others not crucial at all. This implies a need for flexible scheduling options to accommodate individual preferences.

Do you feel that the caregivers are well-trained and equipped to handle your specific needs and preferences?

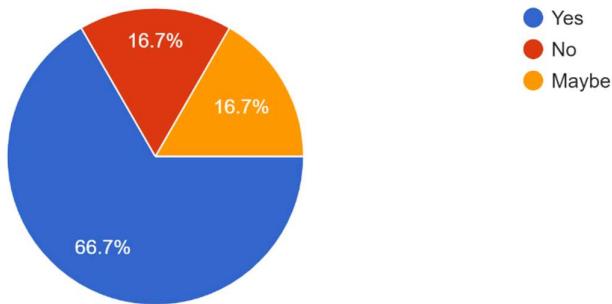
12 responses



Most clients feel that caregivers are well-trained and can handle their specific needs and preferences, but there are instances of dissatisfaction. Continuous training and development initiatives can help in maintaining high competence levels among caregivers.

Would you find value in a feature allowing you to see the caregiver's qualifications, experience, and background checks?

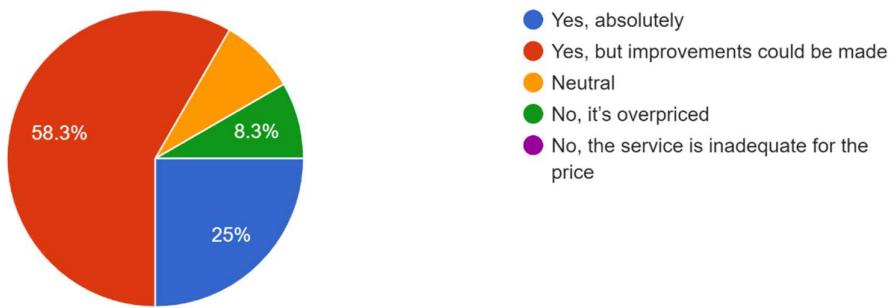
12 responses



The majority find value in a feature allowing them to see the caregiver's qualifications, experience, and background checks. This implies that transparency in caregiver credentials can build trust and confidence among clients.

Do you feel that the services provided are value for money?

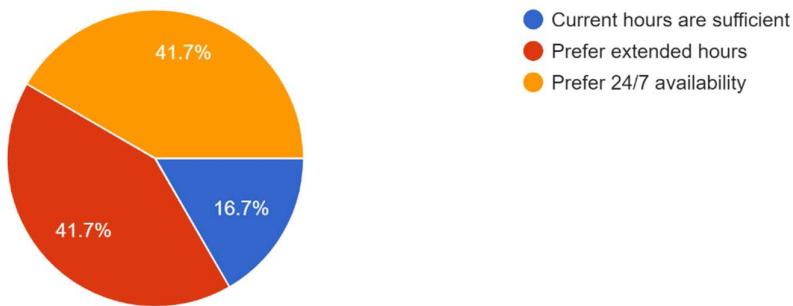
12 responses



While most clients feel the services provided are value for money, there are concerns regarding overpricing. This suggests a need to ensure competitive pricing and clear communication of service value.

Do you find the current service hours sufficient, or would you prefer extended service hours or 24/7 availability?

12 responses



There is a preference for extended service hours or 24/7 availability among clients. This indicates a need for flexible and accommodating service hours to meet diverse client needs.

2.3.3 Requirements Gathered from the Questionnaire:

- A robust mechanism should be embedded to systematically track and analyze caregiver contentment with their working conditions, operational procedures, and client matching process, addressing caregiver concerns about the current system's effectiveness.
- The system must incorporate fields to document the duration and quality of experience each caregiver has in the home care industry, providing insight into their proficiency, expertise, and training needs.

- The database should be capable of logging the types of home care services that each caregiver is willing and able to provide, and it should facilitate efficient service allocation based on individual preferences, skills, and client needs.
- Caregivers should be given the flexibility to choose their preferred means of receiving schedules and updates, and this should align with the client's preferred platforms for providing feedback and reviews about caregivers.
- Enhanced security measures should be in place to ensure secure and streamlined access to clients' medical histories, care plans, and personal details, addressing the caregivers' and clients' needs for confidentiality and comprehensive information.
- The system should integrate advanced features to aid in improving job performance and client matching, based on feedback from both caregivers and clients regarding the effectiveness of the current matching process and the value of services provided.
- A comprehensive assessment of caregivers' comfort and proficiency with technology should be conducted to identify any potential need for additional training and support, ensuring that caregivers are well-equipped to utilize the system effectively.
- A feature to document the proportion of clients requiring personal care assistance is crucial, and it should be optimized based on client feedback regarding the importance of receiving services from the same caregiver each time.
- The system must allow caregivers and clients to efficiently report their activities, concerns, and receive updates, and it should integrate platforms preferred by clients for providing feedback and reviews about caregivers.
- A feature allowing caregivers and clients to evaluate the quality of services, training, and support provided by the agency is essential, enabling continual improvement of support services based on feedback and overall experience ratings.
- Given the preference for digital communication methods, the incorporation of a user-friendly mobile app is essential for managing tasks and receiving updates on the go, catering to the needs of users who find value in such platforms.
- Based on client opinions, robust security measures such as data secrecy, insurance, secure booking processes, and protection against third-party intrusion should be incorporated to protect client information and ensure safety.

- The system should facilitate seamless communication between caregivers and clients and should document the frequency of such interactions to aid in maintaining high levels of client engagement and satisfaction.
- The system development should prioritize features deemed most crucial by caregivers and clients for improving job performance, user satisfaction, and efficiency, ensuring that the system is well-rounded and caters to the needs of both user groups.

2.4 Observations:

2.4.1 Summary of observation/s:

- The system will be used by a diverse range of users, including caregivers, clients, and administrative staff, each with different needs and preferences, necessitating a user-centered design approach.
- Efficient and accurate service allocation is crucial, given the variability in caregiver service preferences, experience levels, and client needs.
- There is a pronounced emphasis on caregiver and client satisfaction, highlighting the need for a user-friendly, intuitive interface and efficient operational workflows within the system.
- The security and confidentiality of client and caregiver information are paramount, necessitating robust security measures and data protection mechanisms.
- The system requires the integration of advanced features such as real-time location tracking, efficient scheduling, and seamless communication to enhance job performance and user experience.
- There is a significant preference for mobile app usage among users for managing tasks and receiving updates, indicating the necessity for a responsive and mobile-friendly design.
- Qualitative feedback from both caregivers and clients is vital for continuous improvement and prioritization of feature development, requiring effective feedback mechanisms and analytical tools within the system.
- The varying levels of technology proficiency among caregivers underscore the importance of incorporating comprehensive training and support evaluation features.

- The diversity in preferred communication channels among users necessitates the incorporation of flexible communication options such as emails, text messages, and in-app notifications.
- The development of features should be prioritized based on the needs and preferences of the end-users, ensuring the system is well-rounded and caters to the needs of both caregivers and clients.
- The database design must facilitate holistic recording of all service interactions, preferences, and feedback to optimize service allocation and enhance user satisfaction.
- Efficient reporting of work hours, activities, and concerns is crucial, requiring streamlined reporting mechanisms and clear communication avenues within the system.
- The diverse user needs, preferences for digital communication, and emphasis on user satisfaction underscore the high importance of user experience design in the development of the system.
- The continuous evaluation and improvement of training, support, and services based on user feedback and experience ratings are essential for maintaining high levels of user satisfaction and system efficiency.

2.4.2 Combined requirements gathered from the observation:

- Develop a user-friendly, intuitive interface to accommodate the diverse user base including caregivers, clients, and administrative staff, ensuring ease of use and accessibility.
- Implement mechanisms for accurate and efficient service allocation, matching caregivers to clients based on service preferences, experience levels, and specific needs.
- Integrate robust security features to protect the confidentiality and integrity of client and caregiver information, addressing privacy concerns and ensuring data security.
- Incorporate advanced features like real-time location tracking and efficient scheduling, enhancing job performance, user experience, and operational efficiency.
- Develop a responsive and mobile-friendly app to cater to the preferences of users who manage tasks and receive updates through mobile platforms, enhancing user experience and accessibility.

- Implement effective feedback mechanisms and analytical tools, facilitating continuous improvement and prioritization of features based on user feedback, ensuring the system meets user needs and expectations.
- Incorporate features to assess and improve the training and support provided to users, identifying areas for enhancement, and ensuring users can effectively utilize the system.
- Implement streamlined reporting mechanisms for efficient reporting of work hours and activities and facilitate clear and effective communication within the system.
- Provide transparent displays of caregivers' qualifications, experience, and background checks, building trust and confidence among clients and ensuring informed decision-making.

2.5 Fact Finding Chart

Objective	Technique	Time Commitment
To get background of the industry and understand the requirements of the project. To provide appropriate allocation of resources.	Background Reading	1.5 days
To scrutinize and learn how a small-scale database is created, optimized, and maintained efficiently.	Case Study	1-2 days
To understand the operational mechanics and service provision methodologies of a home care agency.	Interview (Role Playing)	2-3 hours
To comprehend the challenges encountered by agency owners and identify plausible solutions.	Interview (Role Playing)	2-3 hours
To identify areas that require improvements and enhancements in the existing systems within the industry.	Group Discussion /Brainstorming	4-5 hours
To understand customers' preferences, needs, and expectations from the service providers.	Survey/Questionnaire	30 minutes
To gain insights into the preferences, expectations, and desires of service providers.	Survey/Questionnaire	30 minutes

To ascertain the requisite records and attributes that should be incorporated in the database.	Analysis of Questionnaire	3 hours
To understand the current working environment, operational dynamics, and prevalent business constraints.	Group Discussion	5-6 hours
To discern the software and hardware requirements and constraints pivotal for system development.	Case Study/Interview	6-7 hours

3. List of Requirements:

- Develop advanced algorithms for intelligent matching, considering individual skills, needs, and specializations to ensure optimal assignment and enhanced service quality.
- Implement mechanisms for accurate and efficient service allocation based on service preferences, experience levels, and specific needs.
- Integrate robust security features and stringent measures to safeguard sensitive caregiver and client information, ensuring data security, privacy, and compliance with relevant regulations.
- Develop a user-friendly, intuitive interface and mobile app to accommodate the diverse user base, ensuring ease of use, accessibility, and catering to user preferences for managing tasks and receiving updates through mobile platforms.
- Incorporate detailed reporting features, analytical tools, and formal mechanisms for collecting and analyzing feedback from users, enabling continuous improvement, informed decision-making, and prioritization of feature development.
- Design a robust system for efficient scheduling and conflict resolution, providing real-time updates, instant notifications, and facilitating direct communication between clients and caregivers.
- Maintain exhaustive worker profiles and implement strategies for training and development to balance the distribution of expertise among employees, focusing on areas where expertise is lacking.
- Implement an automated service booking feature integrated with a digital calendar system to manage bookings and allocate service providers accurately.

- Establish policies for providing insurance and safety provisions to employees in high-risk service roles, identifying high-risk roles, and determining the level of coverage to be provided.
- Develop a system for digitizing agency records efficiently, specifying the types of records to be digitized first based on their operational impact.
- Monitor care quality and collect feedback for improvements, focusing on areas that yield the most impact on user satisfaction and efficiency.
- Provide transparent displays of caregivers' qualifications, experience, and background checks to instill trust and confidence among clients.

4. User categories and privileges:

4.1 User Cases:

01. User Name: Admin

Descriptions of Role:

- Configure and maintain the system.
- Manage user accounts and access permissions.
- Update service catalogs and provider information.
- Monitor system usage and performance.
- Address and resolve system issues.

02. User Name: Client

Descriptions of Role:

- Register and manage personal profiles.
- Search and book home care services.
- Review caregiver profiles and service information.
- Communicate with caregivers and the agency.
- Provide feedback on received services.

03. User Name: Caregiver

Descriptions of Role:

- Register and manage professional profiles.
- Update availability and service information.
- Accept and manage service bookings.
- Communicate with clients and the agency.
- Record service details and issues.

04. User Name: Customer Support Representative

Descriptions of Role:

- Address and resolve user issues and concerns.
- Communicate with clients, caregivers, and the admin.
- Monitor user interactions and service delivery.
- Provide information and support to users.
- Record and escalate unresolved issues.

05. User Name: System Analyst**Descriptions of Role:**

- Monitor system performance and user interactions.
- Analyze system data and user feedback.
- Identify and recommend system improvements.
- Ensure system integrity and user satisfaction.
- Report system status and issues to the admin.

4.2 Pre-Post Conditions of User Cases:**1. Client Registration****Description:**

Prospective clients provide personal and, if applicable, medical information to create an account.

Pre-Condition:

The system is set up, configured, and accessible to prospective clients.

Post-Condition:

The client is successfully registered, and their details are securely stored in the system.

2. Service Catalog Management**Description:**

Admin adds, updates, or removes services from the service catalog, ensuring accurate and comprehensive service information is available to users.

Pre-Condition:

The admin is logged in with sufficient permissions.

Post-Condition:

The service catalog is updated with the most current and accurate information.

3. Caregiver Profile Management**Description:**

Caregivers create and manage their professional profiles, including qualifications, services offered, availability, and contact information.

Pre-Condition:

The caregiver is registered and logged in.

Post-Condition:

The caregiver's profile is updated, and the system reflects the most current and accurate information.

4. Service Booking**Description:**

Clients search for services and caregivers, make bookings specifying preferences, and receive booking confirmations.

Pre-Condition:

The client is registered, logged in, and has access to the service catalog and caregiver profiles.

Post-Condition:

The service booking is confirmed, and relevant details are sent to the client and the selected caregiver.

5. Service Delivery and Documentation

Description:

Caregivers provide the booked services and document service details, duration, and any issues encountered.

Pre-Condition:

A service booking is confirmed, and the caregiver has all necessary information and access.

Post-Condition:

The service is provided, documented, and marked as complete or incomplete in the system.

6. Issue Resolution**Description:**

Customer Support Representatives address and resolve user issues, communicating with involved parties and updating the system with resolution details.

Pre-Condition:

A user has reported an issue, and the representative has access to relevant information.

Post-Condition:

The issue is resolved, documented, and the involved parties are informed of the resolution.

7. System Performance Analysis**Description:**

The System Analyst monitors and analyzes system performance, user interactions, and feedback to recommend improvements and optimizations.

Pre-Condition:

The system is operational and generates accurate data and logs.

Post-Condition:

The system is continuously optimized based on analysis findings, ensuring optimal performance and user satisfaction.

8. Feedback Management**Description:**

Clients provide feedback on received services, and the system stores and analyzes the feedback for quality improvement.

Pre-Condition:

A service is completed, and the client is logged in.

Post-Condition:

Client feedback is recorded, analyzed, and used for service quality improvement.

9. Communication Management**Description:**

The system facilitates secure and efficient communication between clients, caregivers, customer support representatives, and the admin.

Pre-Condition:

Users are registered and logged in.

Post-Condition:

Communication is facilitated, recorded, and managed efficiently and securely within the system.

5. List of Assumptions:

- The home care services will primarily be offered within a specific geographic area or set of zip codes, and only residents of these areas can register and request services.
- Each service area or zip code will have an assigned set of caregivers available to provide services, and all service requests within a specific area will be routed to the available caregivers in that area.
- Some interactions between caregivers and clients, such as service delivery and payment collection, might occur offline, and the system will rely on users to update the status and details of such interactions.
- Clients will have the ability to upload a limited number of documents or reports per service request, with a restriction on the total size of the uploaded files.
- A machine learning algorithm may be implemented to assist in optimizing caregiver-client matching, with the understanding that its predictions will improve over time as more data is available, reaching an estimated accuracy of 50-60% after one year.
- In the future, advanced analytics and predictive models might be developed to forecast service demand and optimize resource allocation based on demographic data and service history.
- The system will record service requests with preferred date/time but won't perform detailed analysis to validate caregiver availability or detect scheduling conflicts. Detailed caregiver scheduling and availability management are considered beyond the scope of the current project.
- The system will facilitate online payments through integration with a payment gateway, but the specifics of the payment gateway integration are beyond the scope of the current project.
- It is assumed that clients will have varying preferences and service history, and the system will need to accommodate these variations in service assignments and scheduling.
- The system is assumed to be used by elderly and people with varying levels of technological proficiency. Thus, user interfaces need to be intuitive, and navigational complexity should be minimized.
- It is assumed that the system will need to comply with applicable data protection laws and best practices to ensure the privacy and security of user data.
- The system will be designed to accommodate a diverse and evolving catalog of services, with the flexibility to add, modify, or remove services as needed.

6. Business Constraints:

- a) Secure user authentication is imperative, with role-based access controls ensuring users can only access relevant features and information.

- b) Precise management of each service and its associated caregivers and clients is crucial to ensure accurate service assignments and efficient utilization of resources.
- c) The system must adapt to the evolving service catalog and caregiver qualifications, allowing caregivers the flexibility to offer multiple services.
- d) Every client must have a unique, secure, and easily manageable profile within the system, containing all necessary information and service history.
- e) The system must manage up to 20,000 service requests and bookings per day efficiently, preventing any conflicts and optimizing caregiver schedules.
- f) Compliance with applicable laws and regulations, such as GDPR or HIPAA, is mandatory, ensuring legal adherence in data protection and user privacy.

Noun & Verb Analysis

Nouns: -	Verbs: -
Home care industry	Seek
Caregivers	Prefer
People	Receive
Daily activities	Maintain
Comfort	Turn
Homes	Acknowledges
Service	Want
Individuals	Rely
Elderly	Encounter
Disabilities	Feel
Illness	Expect
Caregiver	Seek
Assistance	Provide
Bathing	Appreciate
Cooking	Ensure
Medication reminders	Handle
Companionship	Appreciate
Management system	Value
Tool	Appreciate
Clients	Appreciate
Expectations	Value
Compassionate care	Appreciate
Physical needs	Appreciate
Emotional support	Look
Reliability	Require
Safety	Expect
Trustworthiness	Rely
Medications	Support
Medical equipment	Expect
Personalization	Educate
Dietary restrictions	Want
Cultural preferences	Take
Care routines	Provide
Effective communication	Manage
Concerns	Deliver
Questions	Offer
Changes	Encourage
Health	Support
Condition	Educate

Responsiveness	Respect
Requests	Encourage
Independence	Face
Expertise	Rely
Medical procedures	Result
Chronic conditions	Need
Rehabilitative care	Disrupt
Dignity	Need
Privacy	Explain
Flexibility	Cause
Circumstances	Experience
Empowerment	Decline
Involvement	Feel
Education	Neglected
Problems	Match
Inadequate Caregiver Match	Connect
Satisfaction	Convey
Well-being	Provide
Care quality	Ask
Care consistency	Address
Client-caregiver relationship	Automate
Misunderstandings	Enhance
Privacy concerns	Streamline
Scheduling	Communicate
Daily routine	Monitor
Anxiety	Protect
Punctuality	Safeguard
Caregiver burnout	Scale
Decline in care quality	Integrate
Neglect	Generate
Home care services	Simplify
High turnover rates	Manage
Client dissatisfaction	Configure
Skilled caregivers	Have access
Recruitment	Provide
Retention	Optimize
Complex regulations	Ensure
Compliance	Integrate
Scheduling complexities	Exchange
Data security	Navigate
Data breaches	Keeping up

Quality assurance	Ensuring
Training	Comply
Performance evaluations	Adhere
Feedback mechanisms	Operates
Insurance coverage	Meets
Liability issues	Protecting
Cost management	Enforces
Technology investments	Safeguards
Digital systems	Scale
Integration	Handle
Healthcare systems	Maintain
Healthcare providers	Ensure
Hospitals	Managing
Primary care physicians	Balancing
Reputational consequences	Implementing
Scalability	Maintaining
Workflow efficiency	Empower
Client information	Answer
Future needs	Keep
Privacy regulations	Expect
Data breach	Encourage
Legal consequences	Support
Financial consequences	Convey
Reputational consequences	Ask
	Respond
	Manage
	Configure
	Provide
	Optimize
	Ensure
	Recognizes
	Commitment
	Revolutionize
	Optimizing
	Enhancing
	Ensuring
	Anticipates

Table.0. All Extracted Nouns & Verbs from Problem Description

<u>Noun:-</u>	<u>Reason :-</u>
Client:	Represents individuals who require home care services.
Caregiver	Represents professionals providing care services.
Service	Describes the type of care services offered.
Schedule	Records scheduling information for caregiver-client assignments.
Messages	Represents communication between clients and caregivers.
Administrator	Represents system administrators overseeing the database
Agency	Represents the home care agency itself.

Table.1. Candidate Entity Set

<u>Verbs :-</u>	<u>Reasons :-</u>
Receive (Client):	Clients receive services from caregivers, making this verb crucial for defining the relationship between clients and caregivers.
Provide (Caregiver, Service):	Caregivers provide services to clients, and services are provided by caregivers. This verb captures the core of the relationship between caregivers and the services they offer.
Manage (Caregiver, Service):	Caregivers manage services, which implies a significant role in organizing and delivering care. This verb defines the responsibilities of caregivers in relation to the services they provide.
Deliver (Caregiver, Service):	Caregivers deliver services to clients, and services are delivered by caregivers. This verb establishes the essential action of delivering care.
Schedule (Client, Administrator, Agency):	Scheduling is a key activity involving clients, administrators (those managing the agency), and the agency itself. Clients schedule care services, administrators schedule caregivers, and agencies manage scheduling.

Communicate (Client, Caregiver, Service):	Communication is essential among clients, caregivers, and services. Clients communicate their needs, caregivers communicate with clients, and services
	communicate information to clients and caregivers.
Coordinate (Administrator, Agency):	Coordination is a fundamental task carried out by administrators within the agency to manage caregivers and clients efficiently. It ensures proper allocation and delivery of care services.
Manage (Administrator, Agency):	Administrators manage the agency, making decisions related to staffing, services, and client assignments.
Engage (Client, Caregiver, Service):	Engagement reflects active involvement, which is crucial for the relationship among clients, caregivers, and the services provided. Clients engage with caregivers and services to receive the care they need.
Administer (Administrator):	Administrators administer the agency's operations, which involves decision-making and oversight, playing a pivotal role in the relationship between the agency and its clients.

Noun	Likely Entity Set to be Assigned
Requirements	Management System
Preferences	Home Care Worker, Client
Skills	Home Care Worker
Quality	Services, Management System
Technologies	Management System
Concerns	Feedback Mechanisms, Home Care Worker, Client
Platform	Management System, Database
Tasks	Functions, Home Care Worker
Updates	Management System, Database, Services
Descriptions	Services, Product's Perspective
Purpose	Management System, Services
Objectives	Management System, Services
Descriptions	Management System, Services
Phase	Transformation, Requirements
Narrative	Management System, Services

Noun	Likely Entity Set to be Assigned
Refinement	Requirements, Services
Audience	Product's Perspective, Management System
State	Transformation, Management System
Consideration	Services, Requirements
Dependencies	Management System, Requirements
Problems	Feedback Mechanisms, Home Care Worker, Client
Depth	Analysis, Consideration
Understanding	Analysis, Consideration
Demographic	Client, Home Care Worker
Experience	Home Care Worker, Client
Duration	Services, Tasks
Specialization	Home Care Worker, Services
Communication Channels	Client, Home Care Worker, Administrative Staff
Training	Home Care Worker, Client
Certifications	Home Care Worker, Services
Billing Details	Client
Payment Information	Client
Shifts	Home Care Worker, Administrative Staff
Schedules	Home Care Worker, Client
Feedback	Client, Home Care Worker, Services
Instructions	Services, Home Care Worker
Care Plans	Client, Services
Safety Provisions	Home Care Worker, Agency
Insurance	Home Care Worker, Client
Records	Database, Agency
Activities	Home Care Worker, Client
Engagement	Client, Home Care Worker
Decisions	Agency, Administrative Staff
Concerns	Client, Home Care Worker

Table 2. Candidate Attributes Set

Noun	Reason for Non-Inclusion
Home care industry	While important, it's a broader concept rather than an entity for relationship definition.
People	General term, not a specific entity.
Daily activities	Abstract concept, not a tangible entity.
Comfort	Abstract concept, not a tangible entity.

Homes	While they are important, they are not entities in the context of an ER diagram.
Individuals	Too general, lacks specificity.
Elderly	May be represented through the "Clients" entity.
Disabilities	Specific cases may be addressed through "Clients" or "Services."
Illness	May be represented through "Clients" or "Services."
Medication reminders	An action or task, not an entity.
Management system	Not a core entity in the context of the Home Care Agency ER diagram.

Tool	Too general and non-specific to the context of the Home Care Agency ER diagram.
Expectations	An abstract concept, not an entity.
Compassionate care	A concept or quality of care, not an entity.
Physical needs	Can be represented through "Clients" or "Services."
Emotional support	Can be represented through "Clients" or "Services."
Trustworthiness	Qualitative attribute, not an entity.
Medications	Can be included as a component within the "Services" entity.
Medical equipment	Can be included as a component within the "Services" entity.
Personalization	Can be included as a feature of "Services" or "Communication."
Dietary restrictions	Can be included as a component within the "Services" entity.
Cultural preferences	Can be included as a component within the "Services" entity.
Care routines	Can be included as a component within the "Services" entity.
Effective communication	Can be included as a feature of "Communication."
Concerns	An abstract concept, not an entity.
Questions	An abstract concept, not an entity.
Changes	An abstract concept, not an entity.
Health	Can be represented through "Clients" or "Services."
Condition	An abstract concept, not an entity.
Responsiveness	Can be included as a feature of "Communication."

Requests	Can be included as a feature of "Communication."
Independence	Can be represented through "Clients."
Expertise	Can be included as a quality of "Caregivers."
Medical procedures	Can be included as a task performed by "Caregivers."
Chronic conditions	Can be included as a condition managed by "Caregivers."
Rehabilitative care	Can be included as a type of "Service" provided by "Caregivers."
Dignity	An abstract concept, not an entity.
Privacy	An abstract concept, not an entity.
Flexibility	An abstract concept, not an entity.
Circumstances	An abstract concept, not an entity.
Empowerment	An abstract concept, not an entity.
Involvement	An abstract concept, not an entity.
Education	An abstract concept, not an entity.
Problems	An abstract concept, not an entity.
Inadequate Caregiver Match	A specific issue, not a core entity for relationship definition.
Satisfaction	An abstract concept, not an entity.
Well-being	Can be represented through "Clients" or "Services."
Care quality	Can be represented through "Services."
Care consistency	Can be represented through "Services."
Client-caregiver relationship	A relationship itself, not a core entity.
Misunderstandings	A specific issue, not a core entity for relationship definition.
Privacy concerns	An abstract concept, not an entity.
Scheduling	Can be represented as a function of "Services" or "Communication."
Daily routine	Can be represented through "Clients" or "Services."
Anxiety	An abstract concept, not an entity.
Punctuality	A specific behaviour, not an entity.
Caregiver burnout	A specific issue, not a core entity for relationship definition.
Decline in care quality	A specific issue, not a core entity for relationship definition.
Neglect	A specific issue, not a core entity for relationship definition.
Home care services	Can be represented through "Services."
High turnover rates	A specific issue, not a core entity for relationship

	definition.
Client dissatisfaction	A specific issue, not a core entity for relationship definition.
Skilled caregivers	Qualitative attribute, not an entity.
Recruitment	A process, not an entity.
Retention	A process, not an entity.
Complex regulations	An abstract concept, not an entity.
Compliance	A concept related to processes, not an entity.
Scheduling complexities	A specific issue, not an entity.
Data security	A concept related to technology and processes, not an entity.
Data breaches	A specific issue, not an entity.
Quality assurance	A concept related to processes, not an entity.
Training	A process, not an entity.
Performance evaluations	A process, not an entity.
Feedback mechanisms	A process, not an entity.
Insurance coverage	An abstract concept, not an entity.
Liability issues	An abstract concept, not an entity.
Cost management	A process, not an entity.
Technology investments	An abstract concept, not an entity.
Digital systems	A concept related to technology, not an entity.
Integration	A concept related to technology, not an entity.
Healthcare systems	An abstract concept, not an entity.
Healthcare providers	An abstract concept, not an entity.
Hospitals	An abstract concept, not an entity.
Primary care physicians	An abstract concept, not an entity.
Reputational consequences	An abstract concept, not an entity.
Scalability	An abstract concept, not an entity.
Workflow efficiency	An abstract concept, not an entity.
Client information	Can be represented through "Clients."
Future needs	An abstract concept, not an entity.
Privacy regulations	An abstract concept, not an entity.

Data breach	A specific issue, not an entity.
Legal consequences	An abstract concept, not an entity.
Financial consequences	An abstract concept, not an entity.
Reputational consequences	An abstract concept, not an entity.

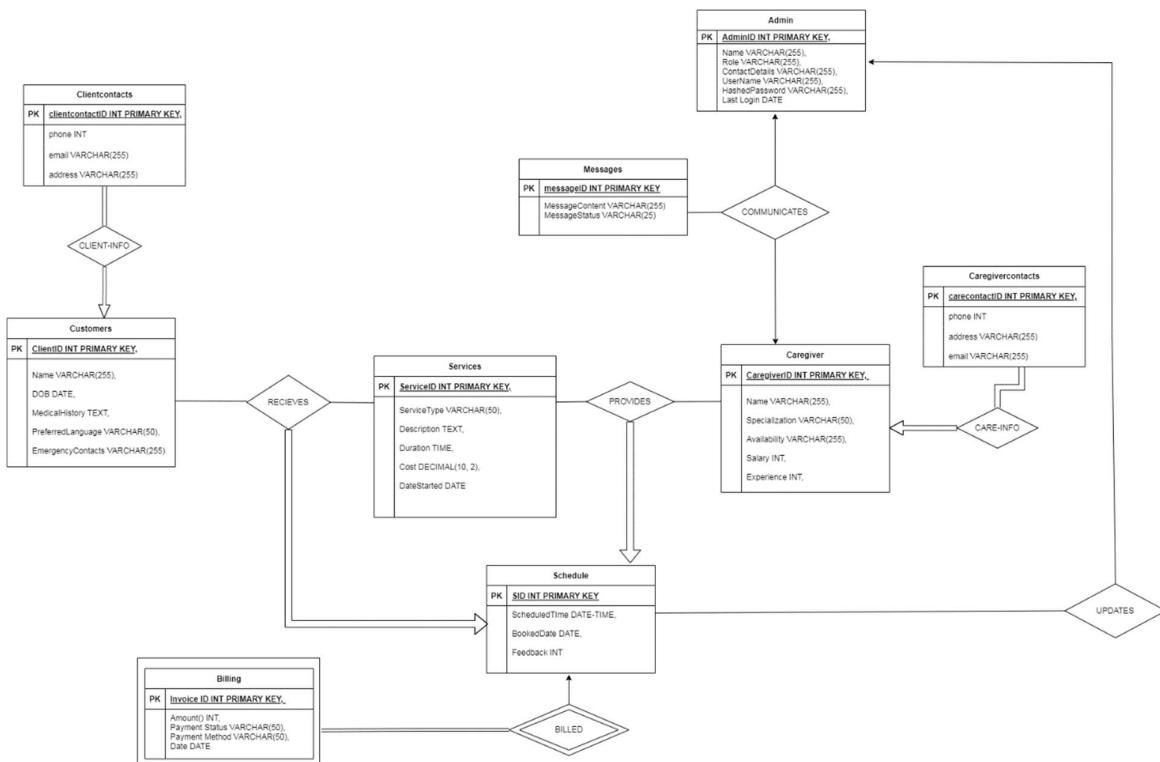
Table.3. Rejected Noun List

Final Entity Set

Client	ClientID INT PRIMARY KEY, Name VARCHAR(255), DateOfBirth DATE, MedicalHistory VARCHAR(255), PreferredLanguage VARCHAR(50), EmergencyContacts VARCHAR(100)
Caregiver	CaregiverID INT PRIMARY KEY, AgencyID INT (Foreign Key), Name VARCHAR(255), Specialization VARCHAR(50), Availability VARCHAR(255), Salary INT, Experience INT,
Service	ServiceID INT PRIMARY KEY, ServiceType VARCHAR(50), Description TEXT, Duration INT, Cost DECIMAL(10, 2),
Schedule	ScheduleID INT PRIMARY KEY, ClientID INT (Foreign key) CaregiverID INT (Foreign Key) ServiceID INT (Foreign key), DateAndTime DATETIME, Status VARCHAR(20)
Messages	MessageID INT PRIMARY KEY, Sender INT (Foreign key to Client, Caregiver, or Administrator), Receiver INT (Foreign key to Client, Caregiver, or Administrator), DateAndTime DATETIME, MessageContent TEXT, MessageStatus VARCHAR(20)
Administrator	AdministratorID INT PRIMARY KEY, Name VARCHAR(255), ContactInformation VARCHAR(255), Role VARCHAR(50), Responsibilities TEXT, Credentials VARCHAR(100), LoginInformation VARCHAR(255)
Billing	Invoice ID INT PRIMARY KEY, Client ID INT (Foreign key to Client), Amount INT, Payment Status VARCHAR (255), Payment Method VARCHAR (30), Date DATETIME

Agency	AgencyID INT PRIMARY KEY, Name VARCHAR(255), ContactInformation VARCHAR(255), Branches VARCHAR(255), Address VARCHAR(255), ServicesOffered TEXT, BillingInformation TEXT
CaregiverContacs	CaregivercontactID INT PRIMARY KEY, Phone INT, Address VARCHAR(255), Email VARCHAR(255)
ClientContacts	clientcontactID INT PRIMARY KEY, Phone INT, Address VARCHAR(255), Email VARCHAR(255)

ERD:



1. Mapping E-R Model to Relational Model:

Client(ClientID, Namse, DOB, MedicalHistory, PreferredLanguage, EmergencyContacts)

ClientContacts(clientcontactID, phone, email, address, ClientID FK)

ClientContacts(clientcontactID, phone, email, address, ClientID FK)

Services(ServiceID, ServiceType, Description, Duration, Cost, DateStarted)

Caregivers(CaregiverID, Name, Specialization, Availability, Salary, Experience)

CaregiverContacts(caregivercontactID, phone, address, email, CaregiverID FK)

Messages(messageID, MessageContent, MessageStatus, ClientID FK, CaregiverID FK, AdminID FK)

Admin/AdminID, Name, Role, ContactDetails, Username, HashedPassword, LastLogin)

Schedule(SID, ClientID FK, ServiceID FK, CaregiverID FK, ScheduledTime, BookedDate)

Billing(InvoiceID, SID FK, Amount, PaymentStatus, PaymentMethod, Date)

2. Schema Refinement Process:

1. First Normal Form (1NF)

Initial Design Considerations:

- Ensure that each table has a primary key that uniquely identifies its rows.
- Eliminate repeating groups or arrays. Ensure that each field contains only atomic values (no lists or sets).

Example Steps to Achieve 1NF:

If a client table has a field for Phones that stores multiple phone numbers, split this into separate records or a separate table.

2. Second Normal Form (2NF)

Prerequisites:

The table is already in 1NF.

Initial Design Considerations:

- Remove partial dependencies, where some non-key attributes depend only on part of a composite primary key.

Example Steps to Achieve 2NF:

If a Schedule table has a composite key (ClientID, ServiceID) and non-key attributes like ClientName and ServiceDescription that depend only on ClientID or ServiceID, respectively, move these to separate Clients or Services tables.

3. Third Normal Form (3NF)

Prerequisites:

The table is already in 2NF.

Initial Design Considerations:

- Remove transitive dependencies, where non-key attributes depend on other non-key attributes.

Example Steps to Achieve 3NF:

If a Billing table includes ClientAddress, which is dependent on ClientID (which in turn is a foreign key from Clients), remove ClientAddress from Billing and ensure it's only in Clients.

4. Boyce-Codd Normal Form (BCNF)

Prerequisites:

The table is already in 3NF.

Initial Design Considerations:

- Remove any remaining anomalies caused by functional dependencies. Every determinant must be a candidate key.

Example Steps to Achieve BCNF:

If a Caregivers table has a CaregiverID and LicenseNumber, where both uniquely identify a caregiver, but LicenseNumber determines CaregiverName, then separate into two tables: one with LicenseNumber and CaregiverName, and another linking LicenseNumber with CaregiverID and other attributes.

3. Normalized Schema:

Client(ClientID, Name, DOB, MedicalHistory, PreferredLanguage, EmergencyContacts)

ClientContacts(clientcontactID, phone, email, address, ClientID FK)

ClientContacts(clientcontactID, phone, email, address, ClientID FK)

Services(ServiceID, ServiceType, Description, Duration, Cost, DateStarted)

Caregivers(CaregiverID, Name, Specialization, Availability, Salary, Experience)

CaregiverContacts(caregivercontactID, phone, address, email, CaregiverID FK)

Messages(messageID, MessageContent, MessageStatus, ClientID FK, CaregiverID FK, AdminID FK)

Admin/AdminID, Name, Role, ContactDetails, Username, HashedPassword, LastLogin)

Schedule(SID, ClientID FK, ServiceID FK, CaregiverID FK, ScheduledTime, BookedDate)

Billing(InvoiceID, SID FK, Amount, PaymentStatus, PaymentMethod, Date)

1. Write DDL Scripts.

-- Customers table

```
CREATE TABLE Client (
```

```
    ClientID INT PRIMARY KEY,
```

```
    Name VARCHAR(255),
```

```
    DOB DATE,
```

```
    MedicalHistory TEXT,
```

```
    PreferredLanguage VARCHAR(50),
```

```
    EmergencyContacts VARCHAR(255)
```

);

-- ClientContacts table

```
CREATE TABLE ClientContacts (
```

```
    clientcontactID INT PRIMARY KEY,
```

```
    phone VARCHAR(255),
```

```
    email VARCHAR(255),
```

```
    address VARCHAR(255),
```

```
    ClientID INT REFERENCES Client(ClientID)
```

);

-- Services table

```
CREATE TABLE Services (
```

```
    ServiceID INT PRIMARY KEY,
```

```
    ServiceType VARCHAR(50),
```

```
    Description VARCHAR(255),
```

```
    Duration VARCHAR(40),
```

```
    Cost DECIMAL(10, 2),
```

```
    DateStarted DATE
```

);

-- Caregivers table

CREATE TABLE Caregivers (

CaregiverID INT PRIMARY KEY,

Name VARCHAR(255),

Availability VARCHAR(255),

Salary VARCHAR(10),

Experience INT

);

-- CaregiverContacts table

CREATE TABLE CaregiverContacts (

caregivercontactID INT PRIMARY KEY,

phone VARCHAR(30),

address VARCHAR(255),

email VARCHAR(255),

CaregiverID INT REFERENCES Caregivers(CaregiverID)

);

-- Admin table

```
CREATE TABLE Admin (
```

```
    AdminID INT PRIMARY KEY,
```

```
    Name VARCHAR(255),
```

```
    Role VARCHAR(255),
```

```
    ContactDetails VARCHAR(255),
```

```
    Username VARCHAR(255),
```

```
    HashedPassword VARCHAR(255),
```

```
    LastLogin DATE
```

```
);
```

-- Messages table

```
CREATE TABLE Messages (
```

```
    messageID INT PRIMARY KEY,
```

```
    MessageContent VARCHAR(200),
```

```
    MessageStatus VARCHAR(50),
```

```
    ClientID INT REFERENCES Client(ClientID),
```

```
    CaregiverID INT REFERENCES Caregivers(CaregiverID),
```

```
AdminID INT REFERENCES Admin (AdminID)
```

```
);
```

```
-- Schedule table
```

```
CREATE TABLE Schedule (
```

```
    SID INT PRIMARY KEY,
```

```
    ClientID INT REFERENCES Client(ClientID),
```

```
    ServiceID INT REFERENCES Services(ServiceID),
```

```
    CaregiverID INT REFERENCES Caregivers(CaregiverID),
```

```
    ScheduledTime VARCHAR(20),
```

```
    BookedDate VARCHAR(25)
```

```
);
```

```
-- Billing table
```

```
CREATE TABLE Billing (
```

```
    InvoiceID INT PRIMARY KEY,
```

```
    SID INT REFERENCES Schedule(SID),
```

```
    Amount DECIMAL(10, 2),
```

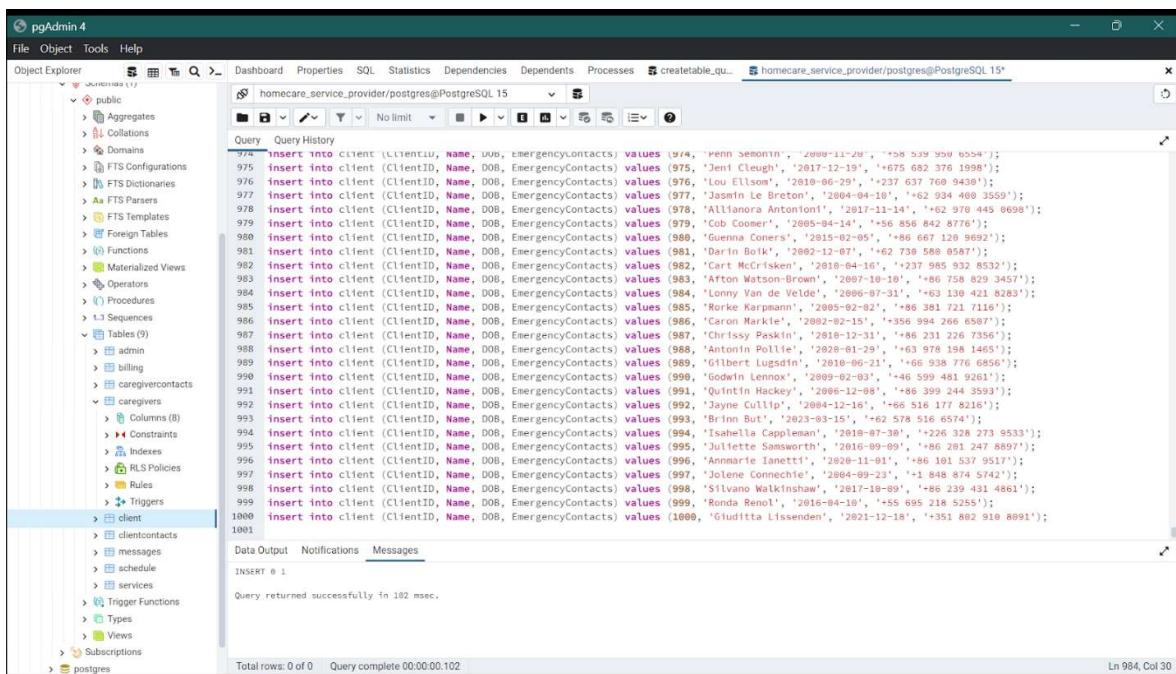
```
    PaymentStatus VARCHAR(50),
```

PaymentMethod VARCHAR(50),

Date VARCHAR(20));

Populating the Tables:

Client Table:



The screenshot shows the pgAdmin 4 interface. The left sidebar (Object Explorer) lists various database objects like public, aggregates, collations, FTS configurations, FTS dictionaries, FTS parsers, FTS templates, foreign tables, functions, materialized views, operators, procedures, sequences, and tables. The 'Tables (9)' section is expanded, showing admin, billing, caregivercontacts, caregivers, client, clientcontacts, messages, schedule, services, trigger functions, types, views, and postgres. The main area (SQL tab) displays a large block of SQL code used to insert data into the 'client' table. The code includes 1000+ insert statements with columns ClientID, Name, DOB, and EmergencyContacts. The query was executed successfully in 102 msec. The status bar at the bottom right indicates 'Ln 984, Col 30'.

```

974   insert into client (ClientID, name, DOB, EmergencyContacts) values (974, 'Penn Semin', '2000-11-20', '+58 539 999 6554');
975   insert into client (ClientID, name, DOB, EmergencyContacts) values (975, 'Jeni Cleugh', '2017-12-19', '+67 682 376 1998');
976   insert into client (ClientID, name, DOB, EmergencyContacts) values (976, 'Lou Ellsom', '2016-06-29', '+237 637 765 9430');
977   insert into client (ClientID, name, DOB, EmergencyContacts) values (977, 'Jasmin Lu Breton', '2004-04-10', '+62 934 400 3559');
978   insert into client (ClientID, name, DOB, EmergencyContacts) values (978, 'Alliamora Antonioni', '2017-11-14', '+62 978 445 0698');
979   insert into client (ClientID, name, DOB, EmergencyContacts) values (979, 'Cob Coomer', '2005-04-14', '+56 856 842 8776');
980   insert into client (ClientID, name, DOB, EmergencyContacts) values (980, 'Guenna Coners', '2015-02-05', '+86 667 126 9692');
981   insert into client (ClientID, name, DOB, EmergencyContacts) values (981, 'Darin Bolk', '2002-12-07', '+62 738 588 0587');
982   insert into client (ClientID, name, DOB, EmergencyContacts) values (982, 'Cart McCraken', '2018-04-16', '+237 985 932 8532');
983   insert into client (ClientID, name, DOB, EmergencyContacts) values (983, 'Afton Watson-Brown', '2007-10-10', '+86 758 829 3457');
984   insert into client (ClientID, name, DOB, EmergencyContacts) values (984, 'Lommy Van de Velde', '2006-07-31', '+63 130 421 8283');
985   insert into client (ClientID, name, DOB, EmergencyContacts) values (985, 'Rorke Karpmann', '2005-02-02', '+86 381 721 7118');
986   insert into client (ClientID, name, DOB, EmergencyContacts) values (986, 'Caron Markle', '2002-02-15', '+356 994 266 6567');
987   insert into client (ClientID, name, DOB, EmergencyContacts) values (987, 'Chrissy Paskin', '2018-12-31', '+86 231 226 7356');
988   insert into client (ClientID, name, DOB, EmergencyContacts) values (988, 'Antonin Pollie', '2028-01-29', '+63 978 198 1465');
989   insert into client (ClientID, name, DOB, EmergencyContacts) values (989, 'Gilbert Lugsdin', '2016-06-21', '+66 938 776 6856');
990   insert into client (ClientID, name, DOB, EmergencyContacts) values (990, 'Godwin Lennox', '2009-02-03', '+46 599 481 9261');
991   insert into client (ClientID, name, DOB, EmergencyContacts) values (991, 'Quintin Hackey', '2006-12-08', '+86 399 244 3593');
992   insert into client (ClientID, name, DOB, EmergencyContacts) values (992, 'Jayne Cullop', '2004-12-16', '+66 516 177 8216');
993   insert into client (ClientID, name, DOB, EmergencyContacts) values (993, 'Brinn But', '2023-03-15', '+62 578 516 6574');
994   insert into client (ClientID, name, DOB, EmergencyContacts) values (994, 'Isabella Cappleman', '2018-07-30', '+226 328 273 9533');
995   insert into client (ClientID, name, DOB, EmergencyContacts) values (995, 'Juliette Sammons', '2016-09-09', '+86 201 247 8897');
996   insert into client (ClientID, name, DOB, EmergencyContacts) values (996, 'Annmarie Janetti', '2020-11-01', '+86 101 537 9517');
997   insert into client (ClientID, name, DOB, EmergencyContacts) values (997, 'Jolene Connechie', '2004-09-23', '+1 848 874 5742');
998   insert into client (ClientID, name, DOB, EmergencyContacts) values (998, 'Silvana Walkinshaw', '2017-10-09', '+86 239 431 4861');
999   insert into client (ClientID, name, DOB, EmergencyContacts) values (999, 'Ronda Renol', '2016-04-18', '+55 695 218 5255');
1000  insert into client (ClientID, name, DOB, EmergencyContacts) values (1000, 'Giuditta Lissenden', '2021-12-18', '+351 802 910 8091");
1001
Data Output Notifications Messages
INSERT 0 1
Query returned successfully in 102 msec.
Total rows: 0 of 0 Query complete 00:00:00.102
Ln 984, Col 30

```

Message Table:

pgAdmin 4

File Object Tools Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes homecare_serv... homecare_service_provider/postgres@PostgreSQL 15*

Query History

```

981 insert into messages (messageid) values (981);
982 insert into messages (messageid) values (982);
983 insert into messages (messageid) values (983);
984 insert into messages (messageid) values (984);
985 insert into messages (messageid) values (985);
986 insert into messages (messageid) values (986);
987 insert into messages (messageid) values (987);
988 insert into messages (messageid) values (988);
989 insert into messages (messageid) values (989);
990 insert into messages (messageid) values (990);
991 insert into messages (messageid) values (991);
992 insert into messages (messageid) values (992);
993 insert into messages (messageid) values (993);
994 insert into messages (messageid) values (994);
995 insert into messages (messageid) values (995);
996 insert into messages (messageid) values (996);
997 insert into messages (messageid) values (997);
998 insert into messages (messageid) values (998);
999 insert into messages (messageid) values (999);
1000 insert into messages (messageid) values (1000);
1001

```

Data Output Notifications Messages

INSERT 0 1

Query returned successfully in 52 msec.

Total rows: 0 of 0 Query complete 00:00:00.052

✓ Query returned successfully in 52 msec. ✎ Ln 1001, Col 28

pgAdmin 4

File Object Tools Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes homecare_serv... homecare_service_provider/postgres@PostgreSQL 15*

Query History

```

1024 SET
1025 messagecontent =
1026 CASE
1027     WHEN random() < 0.1 THEN 'Please remember to submit your weekly reports by Friday.'
1028     WHEN random() < 0.2 THEN 'There's a staff meeting scheduled for Wednesday at 10 AM.'
1029     WHEN random() < 0.3 THEN 'Reminder: The office will be closed next Monday for the holiday.'
1030     WHEN random() < 0.4 THEN 'Completed the daily check-in with all assigned clients.'
1031     WHEN random() < 0.5 THEN 'Requesting time off for the second week of next month.'
1032     WHEN random() < 0.6 THEN 'All medication schedules have been updated as per the latest prescriptions.'
1033     WHEN random() < 0.7 THEN 'Can we reschedule tomorrow's appointment to the afternoon?'
1034     WHEN random() < 0.8 THEN 'I have some questions about the new medication plan.'
1035     ELSE 'Thank you for your assistance today, it was very helpful.'
1036 END,
1037 messagestatus =
1038 CASE
1039     WHEN random() < 0.33 THEN 'sent'
1040     WHEN random() < 0.66 THEN 'received'
1041     ELSE 'read'
1042 END
1043 WHERE messageid BETWEEN 1 AND 1000; -- Adjust this WHERE clause as needed
1044

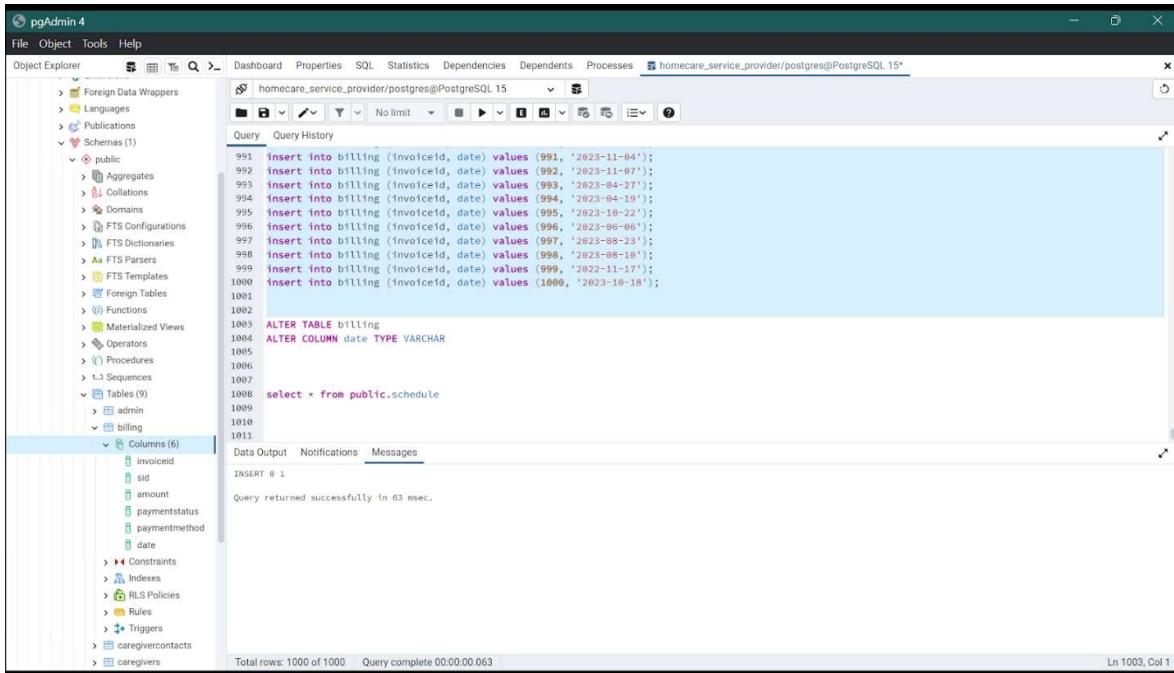
```

UPDATE 1000

Query returned successfully in 54 msec.

Total rows: 0 of 0 Query complete 00:00:00.054

Ln 1051, Col 1

Billing Table:


The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a query editor on the right. The query editor contains the following SQL code:

```

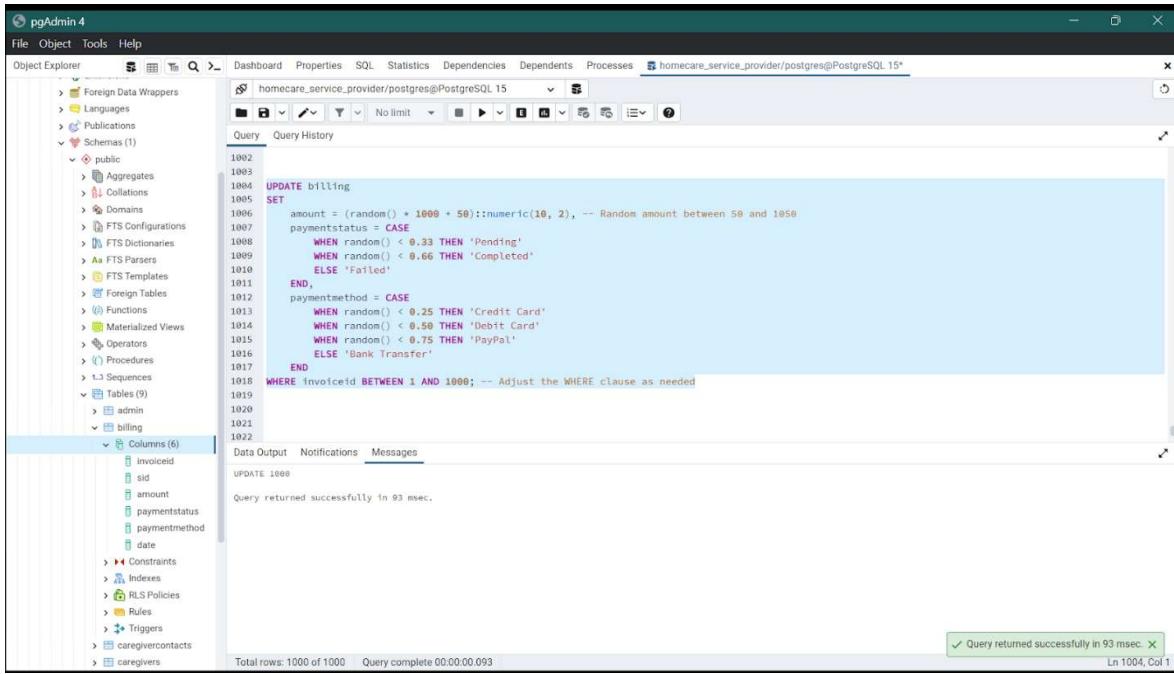
File Object Tools Help
Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes homecare_service_provider/postgres@PostgreSQL 15*
Query Query History
991 insert into billing (invoiceid, date) values (991, '2023-11-04');
992 insert into billing (invoiceid, date) values (992, '2023-11-07');
993 insert into billing (invoiceid, date) values (993, '2023-04-27');
994 insert into billing (invoiceid, date) values (994, '2023-04-19');
995 insert into billing (invoiceid, date) values (995, '2023-18-22');
996 insert into billing (invoiceid, date) values (996, '2023-06-06');
997 insert into billing (invoiceid, date) values (997, '2023-08-23');
998 insert into billing (invoiceid, date) values (998, '2023-08-18');
999 insert into billing (invoiceid, date) values (999, '2022-11-17');
1000 insert into billing (invoiceid, date) values (1000, '2023-10-18');

ALTER TABLE billing
ALTER COLUMN date TYPE VARCHAR

1008 select * from public.schedule
1009
1010
1011

```

The message bar at the bottom indicates "Total rows: 1000 of 1000 Query complete 00:00:00.063".



The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a query editor on the right. The query editor contains the following SQL code:

```

File Object Tools Help
Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes homecare_service_provider/postgres@PostgreSQL 15*
Query Query History
1002
1003
1004 UPDATE billing
1005 SET
1006   amount = (random() * 1000 + 50)::numeric(10, 2), -- Random amount between 50 and 1000
1007   paymentstatus = CASE
1008     WHEN random() < 0.33 THEN 'Pending'
1009     WHEN random() < 0.66 THEN 'Completed'
1010     ELSE 'Failed'
1011   END,
1012   paymentmethod = CASE
1013     WHEN random() < 0.25 THEN 'Credit Card'
1014     WHEN random() < 0.50 THEN 'Debit Card'
1015     WHEN random() < 0.75 THEN 'PayPal'
1016     ELSE 'Bank Transfer'
1017   END
1018 WHERE invoiceid BETWEEN 1 AND 1000; -- Adjust the WHERE clause as needed
1019
1020
1021
1022

```

The message bar at the bottom indicates "Total rows: 1000 of 1000 Query complete 00:00:00.093". A note in the status bar says "✓ Query returned successfully in 93 msec."

Schedule Table:

```

File Object Tools Help
Object Explorer
> FTS Configurations
> FTS Dictionaries
> FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
Tables (9)
> admin
> billing
> caregivercontacts
> caregivers
> client
> clientcontacts
> messages
> schedule
> Columns (6)
  sid
  clientid
  serviceid
  caregiverid
  scheduledtime
  bookeddate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> services
> Types
> Views
> Subscriptions
Query History
984 insert into schedule (sid, scheduledtime, bookeddate) values (984, '8:21 AM', '2023-07-15');
985 insert into schedule (sid, scheduledtime, bookeddate) values (985, '9:12 PM', '2023-02-28');
986 insert into schedule (sid, scheduledtime, bookeddate) values (986, '7:49 PM', '2023-02-10');
987 insert into schedule (sid, scheduledtime, bookeddate) values (987, '8:13 AM', '2023-07-05');
988 insert into schedule (sid, scheduledtime, bookeddate) values (988, '4:43 PM', '2023-01-14');
989 insert into schedule (sid, scheduledtime, bookeddate) values (989, '5:23 PM', '2023-05-18');
990 insert into schedule (sid, scheduledtime, bookeddate) values (990, '10:11 PM', '2023-08-16');
991 insert into schedule (sid, scheduledtime, bookeddate) values (991, '10:59 PM', '2023-02-01');
992 insert into schedule (sid, scheduledtime, bookeddate) values (992, '8:09 PM', '2023-01-05');
993 insert into schedule (sid, scheduledtime, bookeddate) values (993, '3:59 PM', '2023-01-12');
994 insert into schedule (sid, scheduledtime, bookeddate) values (994, '10:14 PM', '2022-12-01');
995 insert into schedule (sid, scheduledtime, bookeddate) values (995, '2:59 PM', '2022-12-30');
996 insert into schedule (sid, scheduledtime, bookeddate) values (996, '7:18 AM', '2023-08-07');
997 insert into schedule (sid, scheduledtime, bookeddate) values (997, '12:59 PM', '2023-02-04');
998 insert into schedule (sid, scheduledtime, bookeddate) values (998, '1:59 PM', '2023-04-02');
999 insert into schedule (sid, scheduledtime, bookeddate) values (999, '9:21 PM', '2023-06-16');
1000 insert into schedule (sid, scheduledtime, bookeddate) values (1000, '4:33 AM', '2023-01-14');

Data Output Notifications Messages
INSERT 0 1
Query returned successfully in 71 msec.

Total rows: 1000 of 1000 Query complete 00:00:00.071
Ln 1007, Col 1

```

Admin Table:

```

File Object Tools Help
Object Explorer
> Aggregates
> Collations
> Domains
> FTS Configurations
> FTS Dictionaries
> FTS Parsers
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
Tables (9)
> admin
> Columns (7)
  adminid
  name
  role
  contactdetails
  username
  hashedpassword
  lastlogin
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> billing
> caregivercontacts
> caregivers
> client
> clientcontacts
> messages
> Columns (6)
  messageneid
Query History
984 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (984, 'Stacia Bass', 'Statistician II', '388 770 647');
985 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (985, 'Jacklyn Dunney', 'VP Sales', '+81 584 851 0159);
986 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (986, 'Clay Ovett', 'Financial Advisor', '+53 174 471);
987 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (987, 'Alfy Phateplace', 'Physical Therapy Assistant');
988 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (988, 'Chere Wasel', 'Developer IV', '+48 524 175 06);
989 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (989, 'Jehanna Ringwood', 'Database Administrator III');
990 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (990, 'Christoper Hyam', 'Data Coordinator', '+38 831 78);
991 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (991, 'YikkI Droogan', 'Account Coordinator', '+82 78);
992 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (992, 'Delphiniti Kehri', 'Research Associate', '+86 5);
993 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (993, 'Baxie Egdal', 'Accountant III', '+7 144 324 81);
994 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (994, 'Alma Bart', 'Geological Engineer', '+84 558 33);
995 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (995, 'Jory Canter', 'Statistician IV', '+46 313 259);
996 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (996, 'Perice Sailor', 'VP Quality Control', '+7 195);
997 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (997, 'Padraig Leathers', 'Structural Analysis Engineer');
998 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (998, 'Roi Serfat', 'Account Representative I', '+34);
999 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (999, 'Minna Gummie', 'Help Desk Technician', '+86 66);
1000 insert into admin (adminid, name, role, contactdetails, username, hashedpassword, lastlogin) values (1000, 'Giff Schwartz', 'Executive Secretary', '+1 334 867 6269);

Data Output Notifications Messages
Total rows: 1000 of 1000 Query complete 00:00:00.108
Ln 1003, Col 1

```

Caregivercontacts Table:

```

977 insert into caregivercontacts (caregivercontactid, phone, address, email) values (977, '+54 133 118 6845', '8 Harbort Place', 'jdinnegrs@biblegateway.com');
978 insert into caregivercontacts (caregivercontactid, phone, address, email) values (978, '+7 918 265 6584', '32 Bayside Park', 'gchokers@evernation.com');
979 insert into caregivercontacts (caregivercontactid, phone, address, email) values (979, '+63 771 826 6671', '38722 Rieder Pass', 'psimkor6t-online.de');
980 insert into caregivercontacts (caregivercontactid, phone, address, email) values (980, '+86 614 628 6533', '7459 Old Gate Street', 'zsentinella7@ft.com');
981 insert into caregivercontacts (caregivercontactid, phone, address, email) values (981, '+358 430 496 6510', '931 Grasskamp Crossing', 'ablemings@cocolo.net');
982 insert into caregivercontacts (caregivercontactid, phone, address, email) values (982, '+57 225 335 4158', '219 Kenwood Drive', 'ggoscombr9@seesaa.net');
983 insert into caregivercontacts (caregivercontactid, phone, address, email) values (983, '+7 497 410 8938', '40 Buhler Pass', 'wenham9@dropbox.com');
984 insert into caregivercontacts (caregivercontactid, phone, address, email) values (984, '+63 566 931 8334', '46943 Huxley Court', 'bradnockrb@ohsu.edu');
985 insert into caregivercontacts (caregivercontactid, phone, address, email) values (985, '+86 248 818 9326', '36 Gerald Circle', 'ashimansc@arizona.edu');
986 insert into caregivercontacts (caregivercontactid, phone, address, email) values (986, '+7 547 375 9887', '744 Gateway Alley', 'tcarlozzird@geocities.com');
987 insert into caregivercontacts (caregivercontactid, phone, address, email) values (987, '+212 328 683 3161', '8799 Heffernan Avenue', 'jzanussifire@ft.com');
988 insert into caregivercontacts (caregivercontactid, phone, address, email) values (988, '+38 837 178 2075', '3 Little Fleur Alley', 'kpambyrfqg.co');
989 insert into caregivercontacts (caregivercontactid, phone, address, email) values (989, '+371 684 762 5514', '5453 Cottonwood Way', 'ahulusrq@izmomo.com');
990 insert into caregivercontacts (caregivercontactid, phone, address, email) values (990, '+48 839 985 1349', '7553 Continental Drive', 'smocherr@cdc.gov');
991 insert into caregivercontacts (caregivercontactid, phone, address, email) values (991, '+62 593 382 7795', '52 Red Cloud Terrace', 'mraper@gravatar.com');
992 insert into caregivercontacts (caregivercontactid, phone, address, email) values (992, '+504 522 199 8889', '237 Killdeer Place', 'kliebemannr@amazonaws.com');
993 insert into caregivercontacts (caregivercontactid, phone, address, email) values (993, '+81 768 585 5281', '48 Reine Alley', 'xflntrkg163.com');
994 insert into caregivercontacts (caregivercontactid, phone, address, email) values (994, '+351 826 853 5357', '7953 Manley Place', 'jmoynham1@cloudflare.com');
995 insert into caregivercontacts (caregivercontactid, phone, address, email) values (995, '+63 988 172 7852', '398 West Road', 'fmaltbyrm@dell.com');
996 insert into caregivercontacts (caregivercontactid, phone, address, email) values (996, '+7 108 576 8513', '44738 Rieder Street', 'awilmutr@latimes.com');
997 insert into caregivercontacts (caregivercontactid, phone, address, email) values (997, '+420 145 690 4597', '9 Dryden Court', 'dmactruramro@moz.org');
998 insert into caregivercontacts (caregivercontactid, phone, address, email) values (998, '+967 863 376 7558', '46276 Gerald Road', 'elambartrpgask.com');
999 insert into caregivercontacts (caregivercontactid, phone, address, email) values (999, '+86 571 431 5031', '8 Sage Way', 'hmckinnonrq@amazon.de');
1000 insert into caregivercontacts (caregivercontactid, phone, address, email) values (1000, '+63 977 591 7195', '993 Helena Parkway', 'bgrindlayr@hostgator.com');

Total rows: 0 of 0   Query complete 00:00:00.075

```

Services Table:

```

975 insert into Services (ServiceID, DateStarted) values (975, '8/10/2023');
976 insert into Services (ServiceID, DateStarted) values (976, '1/17/2023');
977 insert into Services (ServiceID, DateStarted) values (977, '9/36/2023');
978 insert into Services (ServiceID, DateStarted) values (978, '12/10/2022');
979 insert into Services (ServiceID, DateStarted) values (979, '2/6/2023');
980 insert into Services (ServiceID, DateStarted) values (980, '12/25/2022');
981 insert into Services (ServiceID, DateStarted) values (981, '10/1/2023');
982 insert into Services (ServiceID, DateStarted) values (982, '3/7/2023');
983 insert into Services (ServiceID, DateStarted) values (983, '31/28/2022');
984 insert into Services (ServiceID, DateStarted) values (984, '7/1/2023');
985 insert into Services (ServiceID, DateStarted) values (985, '4/28/2023');
986 insert into Services (ServiceID, DateStarted) values (986, '7/3/2023');
987 insert into Services (ServiceID, DateStarted) values (987, '12/18/2022');
988 insert into Services (ServiceID, DateStarted) values (988, '7/22/2023');
989 insert into Services (ServiceID, DateStarted) values (989, '9/21/2023');
990 insert into Services (ServiceID, DateStarted) values (990, '9/26/2023');
991 insert into Services (ServiceID, DateStarted) values (991, '9/27/2023');
992 insert into Services (ServiceID, DateStarted) values (992, '12/17/2022');
993 insert into Services (ServiceID, DateStarted) values (993, '8/4/2023');
994 insert into Services (ServiceID, DateStarted) values (994, '3/8/2023');
995 insert into Services (ServiceID, DateStarted) values (995, '5/28/2023');
996 insert into Services (ServiceID, DateStarted) values (996, '10/29/2023');
997 insert into Services (ServiceID, DateStarted) values (997, '4/26/2023');
998 insert into Services (ServiceID, DateStarted) values (998, '6/14/2023');
999 insert into Services (ServiceID, DateStarted) values (999, '11/22/2022');
1000 insert into Services (ServiceID, DateStarted) values (1000, '9/12/2023');

Total rows: 0 of 0   Query complete 00:00:00.129

```

```

91
92 WHEN servicetype IN ('Cleaning', 'Plumbing', 'Carpentry', 'Appliance Repair', 'Painting') THEN
93 CASE
94     WHEN random() < 0.5 THEN '2 hours'
95     ELSE '4 hours'
96 END
97 ELSE
98 CASE
99     WHEN random() < 0.5 THEN '1 hour'
100    ELSE '2 hours'
101 END
102 WHERE ServiceID BETWEEN 1 AND 1000;
103
104 -- Update the cost
105 UPDATE Services
106 SET
107     cost = (random() * 100 + 20)::numeric(10, 2)
108 WHERE ServiceID BETWEEN 1 AND 1000;
109
110 Data Output Notifications Messages
111 UPDATE 1000
112
113 Query returned successfully in 62 msec.
114
115 Total rows: 1000 of 1000   Query complete 00:00:00.062
116
117 Ln 1, Col 1

```

Caregiver Table:

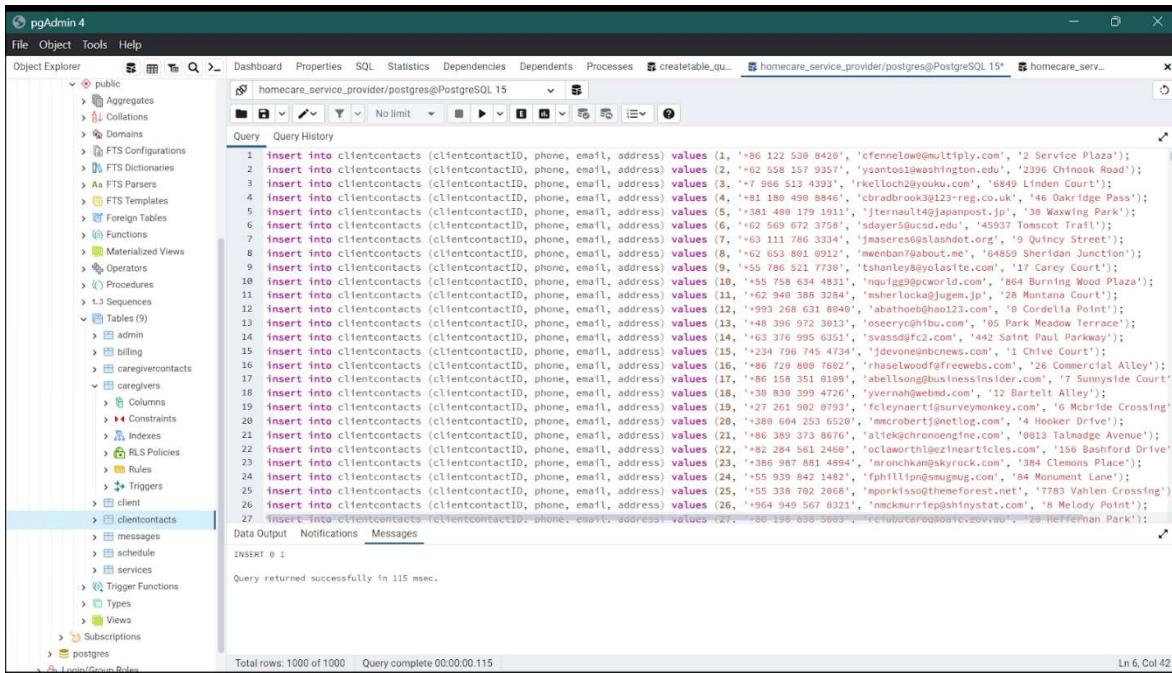
```

983 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (983, 'Fiona Womney', false, '$2511.99', 1);
984 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (984, 'Flynn Wildor', true, '$881.44', 1);
985 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (985, 'Hedvige Kenney', true, '$1971.76', 2);
986 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (986, 'Aldrie Dugall', true, '$1713.45', 4);
987 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (987, 'Korey Abrams', false, '$2676.57', 1);
988 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (988, 'Ileana Unttt', true, '$2287.64', 5);
989 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (989, 'Stace Adshead', true, '$2428.44', 3);
990 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (990, 'Shanda De Launde', true, '$2526.86', 3);
991 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (991, 'Constantin Spens', false, '$2598.19', 4);
992 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (992, 'Dalston Wetherick', false, '$1837.85', 2);
993 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (993, 'Kristos Caswall', false, '$2626.34', 3);
994 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (994, 'Issei Browney', false, '$2249.42', 2);
995 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (995, 'Jacob Fairis', false, '$1719.65', 2);
996 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (996, 'Philippe Louch', true, '$2331.98', 5);
997 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (997, 'Sandrine Eltemere', true, '$868.66', 4);
998 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (998, 'Dabora Celle', true, '$2544.38', 5);
999 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (999, 'Tito Keesman', false, '$2835.33', 5);
1000 Insert into Caregivers (caregiverid, name, availability, salary, experience) values (1000, 'Kerby Beeden', false, '$2881.16', 5);

1001
1002
1003 Data Output Notifications Messages
1004 INSERT 0 1
1005
1006 Query returned successfully in 82 msec.
1007
1008 Total rows: 0 of 0   Query complete 00:00:00.082
1009
1010 Ln 18, Col 37

```

Clientcontacts Table:



```

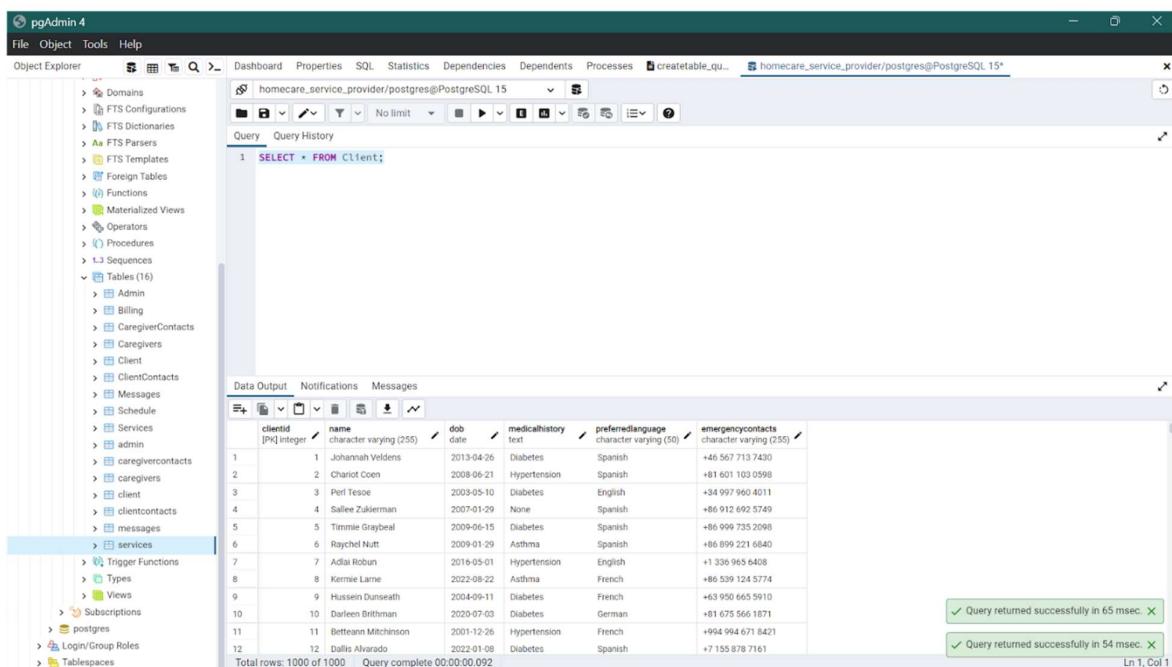
File Object Tools Help
Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes creatable_qu... homecare_service_provider/postgres@PostgreSQL 15* homecare_serv...
Query History
1 insert into clientcontacts (clientcontactID, phone, email, address) values (1, '+86 122 530 8428', 'cfennewlow@multiply.com', '2 Service Plaza');
2 insert into clientcontacts (clientcontactID, phone, email, address) values (2, '+62 558 157 9397', 'ysantosil@washington.edu', '2396 Chinook Road');
3 insert into clientcontacts (clientcontactID, phone, email, address) values (3, '+7 966 513 4393', 'rkelloc2@youku.com', '8849 Linden Court');
4 insert into clientcontacts (clientcontactID, phone, email, address) values (4, '+86 180 499 8846', 'cbradlook9@123-reg.co.uk', '46 Oakridge Pass');
5 insert into clientcontacts (clientcontactID, phone, email, address) values (5, '+831 488 179 1911', 'jternaulat4@apnpost.jp', '38 Maxwing Park');
6 insert into clientcontacts (clientcontactID, phone, email, address) values (6, '+62 569 672 3758', 'sdayer9@uccs.edu', '145937 Tomscot Trail');
7 insert into clientcontacts (clientcontactID, phone, email, address) values (7, '+63 111 786 3334', 'jmaseres9@slashdot.org', '9 Quincy Street');
8 insert into clientcontacts (clientcontactID, phone, email, address) values (8, '+62 852 801 0912', 'mwenban@about.me', '64859 Sheridan Junction');
9 insert into clientcontacts (clientcontactID, phone, email, address) values (9, '+55 786 521 7739', 'tshanley@yolasite.com', '17 Carey Court');
10 insert into clientcontacts (clientcontactID, phone, email, address) values (10, '+55 758 634 4831', 'iniquig9@pcworld.com', '864 Burning Wood Plaza');
11 insert into clientcontacts (clientcontactID, phone, email, address) values (11, '+62 940 581 3284', 'asherlocke@jugem.jp', '28 Montana Court');
12 insert into clientcontacts (clientcontactID, phone, email, address) values (12, '+993 268 631 8840', 'abathoeb@ho123.com', '8 Cordelia Point');
13 insert into clientcontacts (clientcontactID, phone, email, address) values (13, '+48 359 972 3013', 'oseercy@hbu.com', '86 Park Meadow Terrace');
14 insert into clientcontacts (clientcontactID, phone, email, address) values (14, '+63 376 995 6351', 'svassdfc2.com', '442 Saint Paul Parkway');
15 insert into clientcontacts (clientcontactID, phone, email, address) values (15, '+294 796 745 4734', 'jdevone@bcnews.com', '1 Chive Court');
16 insert into clientcontacts (clientcontactID, phone, email, address) values (16, '+86 729 600 7882', 'rhazelwoodff@freewebs.com', '26 Commercial Alley');
17 insert into clientcontacts (clientcontactID, phone, email, address) values (17, '+86 158 351 0189', 'abeltsong@businessinsider.com', '77 Sunnyside Court');
18 insert into clientcontacts (clientcontactID, phone, email, address) values (18, '+36 839 399 4726', 'yvernali@ebnd.com', '12 Bartelt Alley');
19 insert into clientcontacts (clientcontactID, phone, email, address) values (19, '+27 251 982 0793', 'fcleynerf@surveymonkey.com', '6 Mcbride Crossing');
20 insert into clientcontacts (clientcontactID, phone, email, address) values (20, '+386 694 253 6526', 'mrcrobert@netlog.com', '14 Hooker Drive');
21 insert into clientcontacts (clientcontactID, phone, email, address) values (21, '+66 389 373 8676', 'altek@chronengine.com', '10813 Talmadge Avenue');
22 insert into clientcontacts (clientcontactID, phone, email, address) values (22, '+82 284 581 2460', 'octaworth@ezneararticles.com', '156 Bashford Drive');
23 insert into clientcontacts (clientcontactID, phone, email, address) values (23, '+366 987 881 4894', 'ironchamg@skyrock.com', '1384 Clemens Place');
24 insert into clientcontacts (clientcontactID, phone, email, address) values (24, '+65 939 642 1482', 'fpittifng@smugug.com', '84 Monument Lane');
25 insert into clientcontacts (clientcontactID, phone, email, address) values (25, '+55 338 782 2664', 'imorkisso@themeforest.net', '7783 Vahlen Crossing');
26 insert into clientcontacts (clientcontactID, phone, email, address) values (26, '+964 949 567 0321', 'nackmurrtep@shinystat.com', '8 Melody Point');
27 INSERT INTO clientcontacts (clientcontactID, phone, email, address) VALUES (27, '+86 158 650 5693', 'reclubutor@osobito20v1ua... 20 Heffelhan Park');

Data Output Notifications Messages
Total rows: 1000 of 1000 Query complete 00:00:00.115 Ln 6, Col 42

```

2. SQL Queries

1. Simple Select: Retrieve all records from the Customers table.



```

File Object Tools Help
Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes creatable_qu... homecare_service_provider/postgres@PostgreSQL 15* homecare_serv...
Query History
1 SELECT * FROM Client;

Data Output Notifications Messages
clientid [PK] integer
name character varying(255)
dob date
medicalhistory text
preferredlanguage character varying(10)
emergencycontacts character varying(250)

1 1 Johannah Veldens 2013-04-26 Diabetes Spanish +46 567 713 7430
2 2 Chariot Coen 2008-06-21 Hypertension Spanish +81 601 103 0598
3 3 Perl Tesoe 2003-05-10 Diabetes English +34 997 960 4011
4 4 Sallez Zukierman 2007-01-29 None Spanish +86 912 692 5749
5 5 Timmie Graybeal 2009-06-15 Diabetes Spanish +86 999 735 2096
6 6 Raychel Nut 2009-01-29 Asthma Spanish +46 899 221 6840
7 7 Adial Robun 2016-05-01 Hypertension English +1 336 965 6408
8 8 Kerrie Larne 2022-08-22 Asthma French +86 539 124 5774
9 9 Hussein Dunseath 2004-09-11 Diabetes French +63 950 665 5910
10 10 Darleen Brithman 2020-07-02 Diabetes German +81 675 566 1871
11 11 Bettiean Mitchinson 2001-12-26 Hypertension French +94 994 671 8421
12 12 Dallis Avardao 2022-01-08 Diabetes Spanish +7 155 878 7161

Total rows: 1000 of 1000 Query complete 00:00:00.092 Ln 1, Col 1

```

SELECT * FROM Customers;

1. Filtering: Retrieve customers who have 'Asthma' in their medical history.
DAIICT, Gandhinagar, Gujarat

2023

Ans :-

The screenshot shows the pgAdmin 4 interface with a query window displaying the results of the following SQL statement:

```
1 SELECT * FROM Client WHERE MedicalHistory = 'Asthma';
```

The results table contains 12 rows of customer data, each with columns: clientid, name, dob, medicalhistory, preferredlanguage, and emergencycontacts.

clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts
1	Raychel Nutt	2009-01-29	Asthma	Spanish	+86 899 221 6840
2	Kermie Larne	2022-08-22	Asthma	French	+86 539 124 5774
3	Antonio Taggart	2022-01-29	Asthma	Spanish	+86 730 317 5428
4	Cori Skirvin	2021-09-29	Asthma	French	+46 690 114 2612
5	Perceval Collaton	2021-03-17	Asthma	German	+33 345 151 3592
6	Darby Simonsen	2013-08-23	Asthma	Chinese	+63 791 281 7642
7	Barbara-anne Neathway	2005-03-15	Asthma	Spanish	+86 357 691 2938
8	Dav Gillison	2014-10-07	Asthma	French	+7 109 463 1070
9	Lev Plumstead	2016-03-15	Asthma	Spanish	+46 519 615 6507
10	Gardner Felderer	2002-06-13	Asthma	Spanish	+58 207 125 5889
11	Crista Owenson	2012-02-07	Asthma	English	+255 848 978 9569
12	Beale Worling	2012-07-31	Asthma	Spanish	+1 538 991 0808

Total rows: 225 of 225 Query complete 00:00:00.059

SELECT * FROM Customers WHERE MedicalHistory = 'Asthma';

2. Sorting: Retrieve customers sorted by their date of birth in ascending order.

ANS :-

The screenshot shows the pgAdmin 4 interface with a query window displaying the results of the following SQL statement:

```
1 SELECT * FROM Client ORDER BY DOB ASC;
```

The results table contains 12 rows of customer data, each with columns: clientid, name, dob, medicalhistory, preferredlanguage, and emergencycontacts.

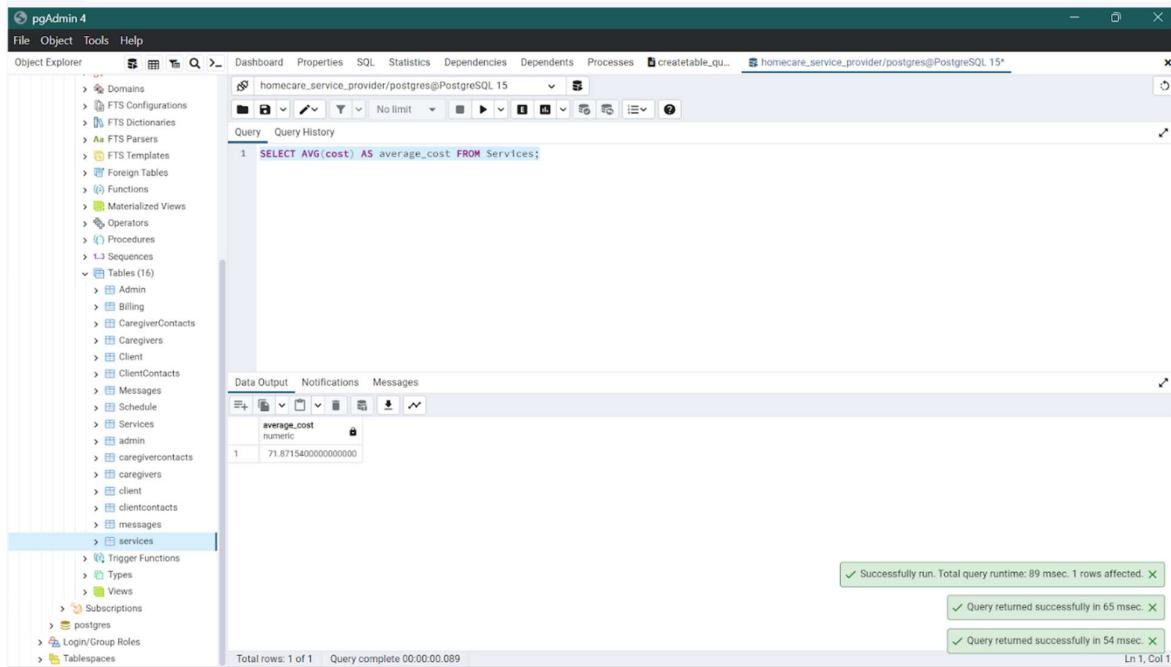
clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts
1	Julissa Plumbridge	2000-11-18	Asthma	German	+351 773 357 1326
2	Penn Seminon	2000-11-20	Hypertension	French	+50 539 950 6584
3	Rowan McVey	2000-12-07	None	Spanish	+50 999 686 7570
4	Fabien Yukhin	2000-12-25	Diabetes	Spanish	+351 595 953 2240
5	Sorcha Eylek	2000-12-29	Diabetes	Spanish	+1 206 424 8276
6	Wood Pacey	2001-01-06	Asthma	German	+30 103 330 9647
7	Roy Artois	2001-01-28	Hypertension	German	+1 974 464 3160
8	Maria Isaac	2001-01-31	No known allergies	Spanish	+33 187 648 3133
9	Austin Mattys	2001-02-05	Asthma	Spanish	+61 815 738 2933
10	Nitsy Peller	2001-02-05	Diabetes	English	+48 249 914 4330
11	Maricelle MacGorman	2001-02-13	Hypertension	English	+57 472 704 8367
12	Harrison Pierton	2001-02-15	Hypertension	English	+48 174 591 9877

Total rows: 1000 of 1000 Query complete 00:00:00.115

SELECT * FROM Customers ORDER BY DOB ASC;

3. Aggregation: Calculate the average cost of services from the Services table

ANS:-



```
pgAdmin 4
File Object Tools Help
Object Explorer   Dashboard Properties SQL Statistics Dependencies Dependents Processes creatable_qu... homecare_service_provider/postgres@PostgreSQL 15*
Query Query History
1 SELECT AVG(cost) AS average_cost FROM Services;
Data Output Notifications Messages
average_cost
1 71.87154000000000
Total rows: 1 of 1  Query complete 00:00:00.089
Ln 1, Col 1
```

Successfully run. Total query runtime: 89 msec. 1 rows affected.

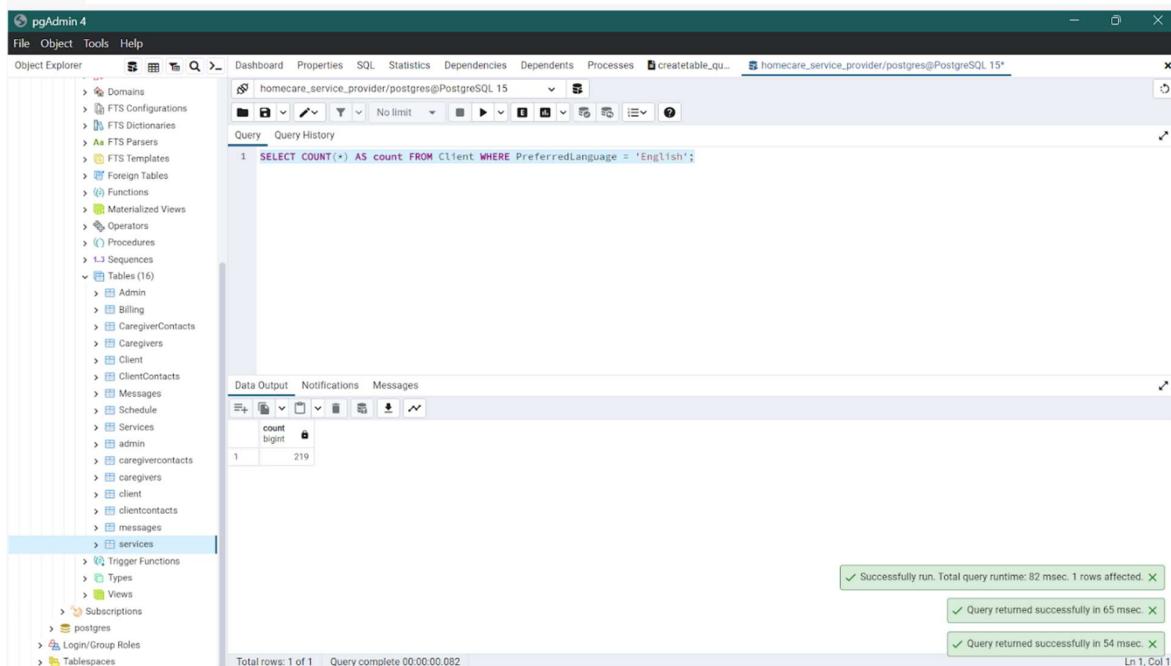
Query returned successfully in 65 msec.

Query returned successfully in 54 msec.

`SELECT AVG(cost) AS average_cost FROM Services;`

4. Counting: Count the number of customers who prefer 'French' as their language.

ANS:-



```
pgAdmin 4
File Object Tools Help
Object Explorer   Dashboard Properties SQL Statistics Dependencies Dependents Processes creatable_qu... homecare_service_provider/postgres@PostgreSQL 15*
Query Query History
1 SELECT COUNT(*) AS count FROM Client WHERE PreferredLanguage = 'English';
Data Output Notifications Messages
count
1 219
Total rows: 1 of 1  Query complete 00:00:00.082
Ln 1, Col 1
```

Successfully run. Total query runtime: 82 msec. 1 rows affected.

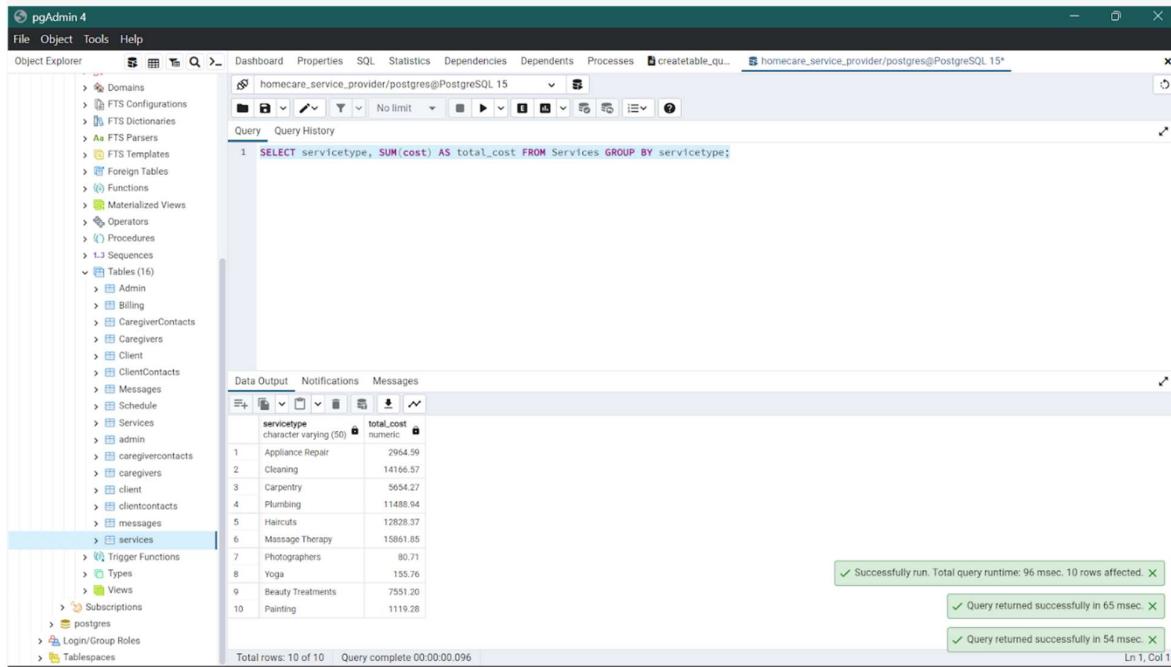
Query returned successfully in 65 msec.

Query returned successfully in 54 msec.

`SELECT COUNT(*) AS count FROM Customers WHERE PreferredLanguage = 'Hindi';`

5. Grouping: Group services by service type and calculate the total cost for each type.

ANS :-



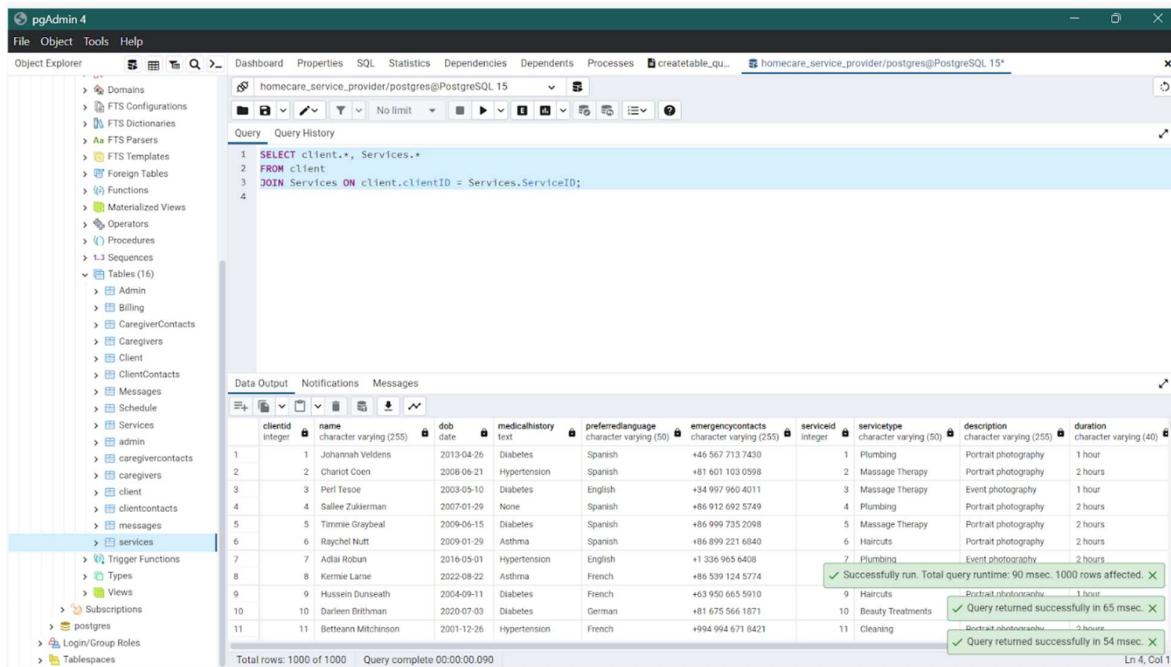
```
1 SELECT servicetype, SUM(cost) AS total_cost FROM Services GROUP BY servicetype;
```

servicetype	total_cost
Appliance Repair	2964.59
Cleaning	14166.57
Carpentry	5654.27
Plumbing	11488.94
Haircut	12828.37
Massage Therapy	15801.85
Photographers	80.71
Yoga	153.76
Beauty Treatments	7551.20
Painting	1119.28

SELECT servicetype, SUM(cost) AS total_cost FROM Services GROUP BY servicetype;

6. Join: Retrieve a list of customers along with the services they have used.

ANS :-



```
1 SELECT client.*, Services.*  
2 FROM client  
3 JOIN Services ON client.clientID = Services.ServiceID;  
4
```

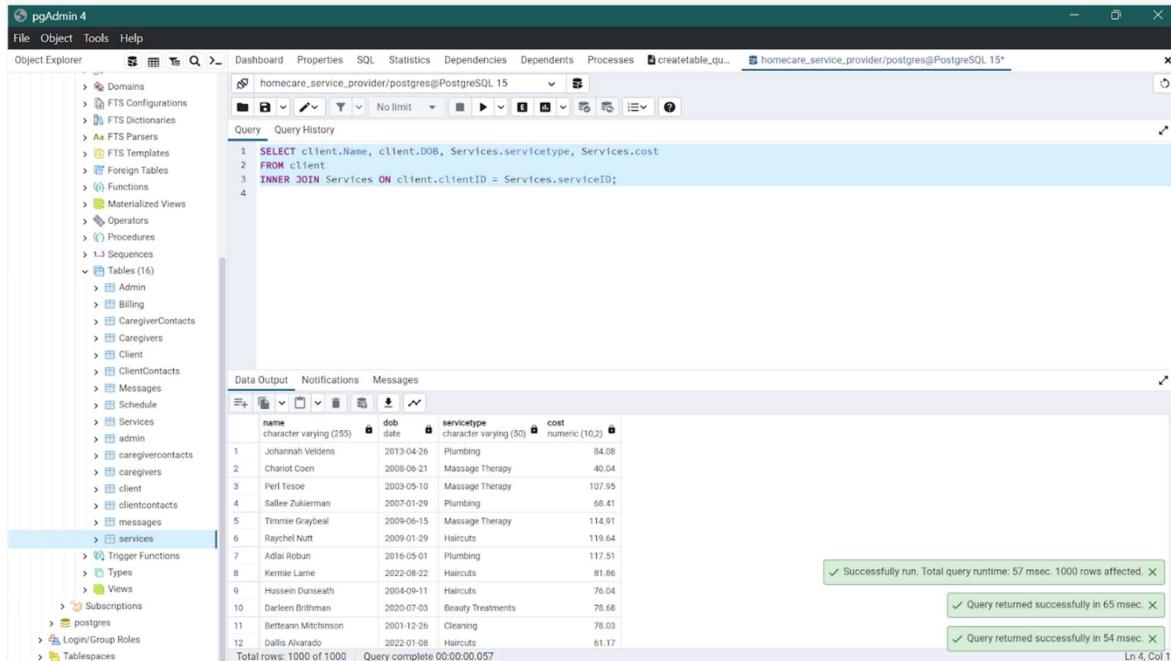
clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts	serviceid	servicetype	description	duration
1	Johannah Veldens	2013-04-26	Diabetes	Spanish	+46 567 713 7430	1	Plumbing	Portrait photography	1 hour
2	Charlot Coen	2008-06-21	Hypertension	Spanish	+81 601 103 0598	2	Massage Therapy	Portrait photography	2 hours
3	Perf Tesoe	2003-05-10	Diabetes	English	+34 997 960 4011	3	Massage Therapy	Event photography	1 hour
4	Salleve Zukerman	2007-01-29	None	Spanish	+86 912 692 5749	4	Plumbing	Portrait photography	2 hours
5	Timmie Graybeal	2009-06-15	Diabetes	Spanish	+86 999 735 2088	5	Massage Therapy	Portrait photography	2 hours
6	Raychel Nutt	2009-01-29	Asthma	Spanish	+86 899 221 6840	6	Haircuts	Portrait photography	2 hours
7	Adil Roban	2016-05-01	Hypertension	English	+1 336 965 6408	7	Plumbing	Event photography	2 hours
8	Kermie Lame	2022-08-22	Asthma	French	+86 539 124 5774	8	Portrait,photography	Portrait,photography	1 hour
9	Hussein Dunseath	2004-09-11	Diabetes	French	+63 950 665 5910	9	Haircuts	Portrait,photography	1 hour
10	Darleen Brithman	2020-07-03	Diabetes	German	+86 675 566 1871	10	Beauty Treatments	Portrait,photography	2 hours
11	Bettieann Mitchinson	2001-12-26	Hypertension	French	+994 994 671 8421	11	Cleaning	Portrait,photography	2 hours

SELECT Customers.* , Services.*
FROM Customers

JOIN Services ON Customers.CustomerID = Services.CustomerID;

7. Inner Join: Retrieve customers who have used services and display their details along with the service details.

ANS :-



```

SELECT client.Name, client.DOB, Services.servicetype, Services.cost
FROM client
INNER JOIN Services ON client.clientID = Services.serviceID;

```

	name	dob	servicetype	cost
1	Johannah Veldens	2013-04-26	Plumbing	84.08
2	Chariot Coen	2008-06-21	Massage Therapy	40.04
3	Pet Tesoe	2003-05-10	Massage Therapy	107.95
4	Sallee Zukierman	2007-01-29	Plumbing	68.41
5	Timmie Graybeal	2009-06-15	Massage Therapy	114.91
6	Raychel Nutt	2009-01-29	Haircuts	119.64
7	Adai Robin	2016-05-01	Plumbing	117.51
8	Kermie Larne	2022-08-22	Haircuts	81.86
9	Hussein Dunseath	2004-09-11	Haircuts	76.04
10	Darleen Britzman	2020-07-03	Beauty Treatments	78.68
11	Betteann Mitchinson	2001-12-26	Cleaning	78.03
12	Dallai Alvarado	2022-01-08	Haircuts	61.17

Successfully run. Total query runtime: 57 msec. 1000 rows affected.

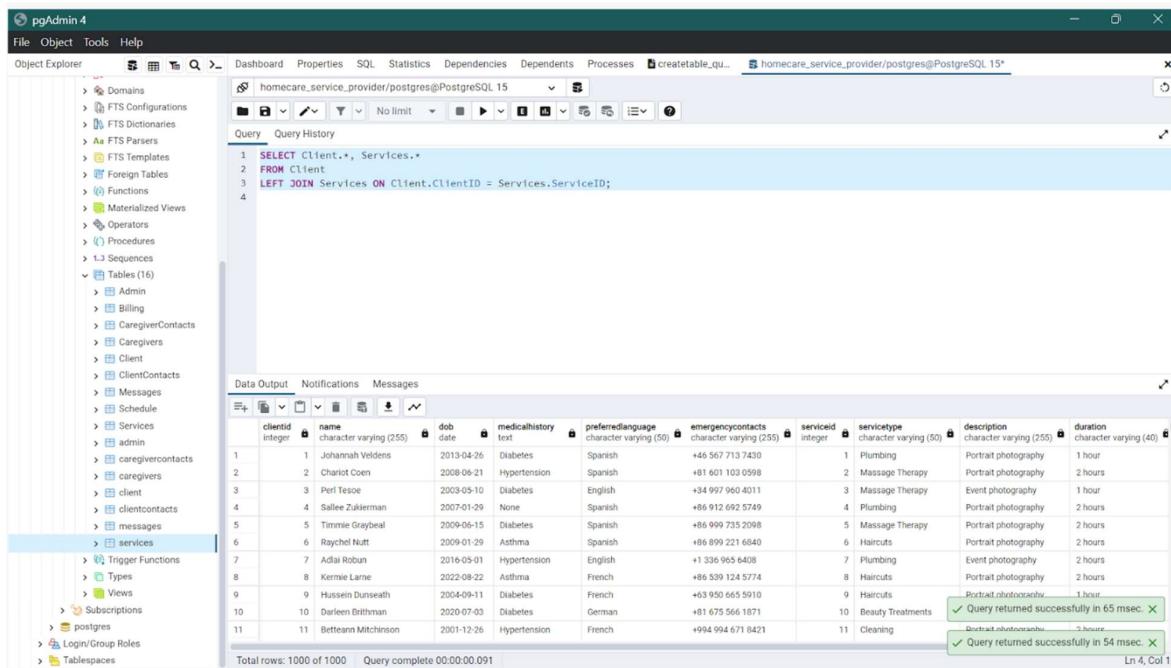
Query returned successfully in 65 msec.

Query returned successfully in 54 msec.

SELECT Customers.Name, Customers.DOB, Services.servicetype, Services.cost
 FROM Customers
 INNER JOIN Services ON Customers.CustomerID = Services.CustomerID;

8. Left Join: Retrieve all customers and their services, including those who haven't used any services.

ANS :-



```

SELECT Client.* , Services.*
FROM Client
LEFT JOIN Services ON Client.ClientID = Services.CustomerID;

```

clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts	serviceid	servicetype	description	duration
1	Johannah Veldens	2013-04-26	Diabetes	Spanish	+46 567 713 7430	1	Plumbing	Portrait photography	1 hour
2	Charlot Coen	2008-06-21	Hypertension	Spanish	+81 601 103 0598	2	Massage Therapy	Portrait photography	2 hours
3	Perrt Tesoe	2003-05-10	Diabetes	English	+34 997 960 4011	3	Massage Therapy	Event photography	1 hour
4	Sallee Zukierman	2007-01-29	None	Spanish	+86 912 692 5749	4	Plumbing	Portrait photography	2 hours
5	Timmie Graybeal	2009-06-15	Diabetes	Spanish	+86 999 735 2098	5	Massage Therapy	Portrait photography	2 hours
6	Ryorchel Nutt	2009-01-29	Asthma	Spanish	+86 899 221 6840	6	Haircuts	Portrait photography	2 hours
7	Adilai Robun	2016-05-01	Hypertension	English	+1 336 965 6408	7	Plumbing	Event photography	2 hours
8	Kermie Lame	2022-08-22	Asthma	French	+86 539 124 5774	8	Haircuts	Portrait photography	2 hours
9	Hussein Dunseath	2004-09-11	Diabetes	French	+63 950 665 5910	9	Haircuts	Portrait photography	1 hour
10	Darleen Brithman	2020-07-03	Diabetes	German	+81 675 566 1871	10	Beauty Treatments	Portrait photography	2 hours
11	Betteann Mitchinson	2001-12-26	Hypertension	French	+994 994 671 8421	11	Cleaning	Portrait photography	2 hours

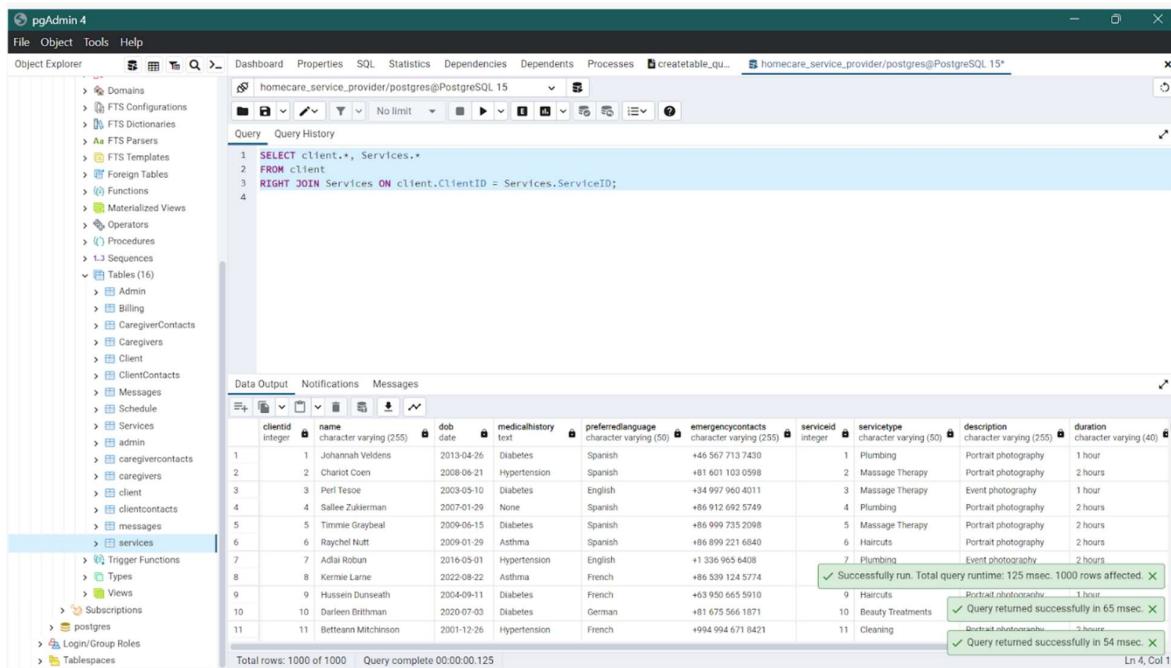
```

SELECT Client.* , Services.*
FROM Client
LEFT JOIN Services ON Client.ClientID = Services.CustomerID;

```

9. Right Join: Retrieve all services and the customers who have used them, including services without customers.

ANS :-



```

SELECT client.* , Services.*
FROM client
RIGHT JOIN Services ON client.ClientID = Services.ServiceID;

```

	clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts	serviceid	servicetype	description	duration
1	1	Johannah Veldens	2013-04-26	Diabetes	Spanish	+46 567 713 7430	1	Plumbing	Portrait photography	1 hour
2	2	Charlot Coen	2008-06-21	Hypertension	Spanish	+81 601 103 0598	2	Massage Therapy	Portrait photography	2 hours
3	3	Perrt Tesoe	2003-05-10	Diabetes	English	+34 997 960 4011	3	Massage Therapy	Event photography	1 hour
4	4	Sallee Zukierman	2007-01-29	None	Spanish	+86 912 692 5749	4	Plumbing	Portrait photography	2 hours
5	5	Timmie Graybeal	2009-06-15	Diabetes	Spanish	+86 999 735 2098	5	Massage Therapy	Portrait photography	2 hours
6	6	Ryorchel Nutt	2009-01-29	Asthma	Spanish	+86 899 221 6840	6	Haircuts	Portrait photography	2 hours
7	7	Adilai Robun	2016-05-01	Hypertension	English	+1 336 965 6408	7	Plumbing	Event photography	2 hours
8	8	Kermie Lame	2022-08-22	Asthma	French	+86 539 124 5774	9	Haircuts	Portrait photography	1 hour
9	9	Hussein Dunseath	2004-09-11	Diabetes	French	+63 950 665 5910	10	Beauty Treatments	Query returned successfully in 65 msec.	0 hours
10	10	Darleen Brithman	2020-07-03	Diabetes	German	+81 675 566 1871	11	Cleaning	Portrait photography	54 msec.
11	11	Betteann Mitchinson	2001-12-26	Hypertension	French	+994 994 671 8421				

```

SELECT Customers.* , Services.*
FROM Customers
RIGHT JOIN Services ON Customers.CustomerID = Services.CustomerID;

```

10. Subquery: Find customers who have used services with a cost greater than the average cost.

ANS :-

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and several tables under the 'Tables (16)' section. The right pane is the Query Editor, showing a query and its execution results.

```

SELECT * FROM client
WHERE ClientID IN (SELECT ClientID FROM Services WHERE cost > (SELECT AVG(cost) FROM Services));

```

The results table shows 12 rows of data from the 'client' table:

clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts
1	Johannah Veldens	2013-04-26	Diabetes	Spanish	+46 567 713 7430
2	Charict Coen	2008-06-21	Hypertension	Spanish	+81 601 103 0598
3	Perf Tesoe	2003-05-10	Diabetes	English	+34 997 960 4011
4	Sallez Zukerman	2007-01-29	None	Spanish	+61 912 692 5749
5	Timmie Graybeal	2009-06-15	Diabetes	Spanish	+61 999 735 2096
6	Rachel Nutt	2009-01-29	Asthma	Spanish	+61 899 221 6840
7	Adlai Robun	2016-05-01	Hypertension	English	+1 336 965 6408
8	Kermie Larne	2022-08-22	Asthma	French	+61 539 124 5774
9	Hussein Dunseath	2004-09-11	Diabetes	French	+63 950 665 5910
10	Darleen Brithman	2020-07-03	Diabetes	German	+61 675 566 1871
11	Betteann Mitchinson	2001-12-26	Hypertension	French	+994 994 671 8421
12	Dallas Alvarado	2022-01-08	Diabetes	Spanish	+7 155 878 7161

Total rows: 1000 of 1000 Query complete 00:00:00.104 Lr 3, Col 1

SELECT * FROM Customers
WHERE CustomerID IN (SELECT CustomerID FROM Services WHERE cost > (SELECT AVG(cost) FROM Services));

11. IN Operator: Retrieve customers who prefer either 'English' or 'Hindi'.

ANS :-

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows a tree view of database objects including Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16). The 'Tables' node is expanded.
- Query Editor:** Contains the SQL query: `SELECT * FROM Client WHERE PreferredLanguage IN ('English', 'Spanish');`
- Data Output:** Displays the results of the query in a table format. The columns are: clientid (PK integer), name (character varying(255)), dob (date), medicalhistory (text), preferredlanguage (character varying(50)), and emergencycontacts (character varying(255)). The results show 12 rows of client information, all of whom prefer English or Spanish.
- Messages:** Shows three success messages indicating the query was run successfully and returned results in 65 msec and 54 msec respectively.

```
SELECT * FROM Client WHERE PreferredLanguage IN ('English', 'Spanish');
```

12. NOT IN Operator: Retrieve customers who don't prefer 'English' as their language.

ANS :-

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows a tree view of database objects including Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16). The 'Tables' node is expanded.
- Query Editor:** Contains the SQL query: `SELECT * FROM Client WHERE PreferredLanguage NOT IN ('English');`
- Data Output:** Displays the results of the query in a table format. The columns are: clientid (PK integer), name (character varying(255)), dob (date), medicalhistory (text), preferredlanguage (character varying(50)), and emergencycontacts (character varying(255)). The results show 781 rows of client information, all of whom prefer a language other than English.
- Messages:** Shows three success messages indicating the query was run successfully and returned results in 65 msec and 54 msec respectively.

SELECT * FROM Customers WHERE PreferredLanguage NOT IN ('English');

13. Between Operator: Find customers born between '1990-01-01' and '2000-12-31'.

ANS :-

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows a tree view of database objects including Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16).
- Query Editor:** Contains the SQL query: `SELECT * FROM Client WHERE DOB BETWEEN '1990-01-01' AND '2000-12-31';`
- Data Output:** Displays a table with 5 rows of customer data:

clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts
1	Rowan McVey	2000-12-07	None	Spanish	+55 999 666 7570
2	Julissa Plumbridge	2000-11-18	Asthma	German	+351 773 357 1326
3	Fabien Yukhin	2000-12-25	Diabetes	Spanish	+351 595 933 2240
4	Sorcha Eylel	2000-12-29	Diabetes	Spanish	+1 206 424 8276
5	Penin Semonin	2000-11-20	Hypertension	French	+58 539 950 6554

- Messages:** Shows three success messages indicating the query was run successfully and returned results in 60 msec, 65 msec, and 54 msec respectively.

SELECT * FROM Customers WHERE DOB BETWEEN '1990-01-01' AND '2000-12-31';

14. Like Operator: Search for customers whose names start with 'J'.

ANS :-

```
1 SELECT * FROM Client WHERE Name LIKE 'J%';
```

clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts
1	Johannah Veldens	2013-04-26	Diabetes	Spanish	+46 567 713 7430
2	Jess Caskey	2005-06-21	Diabetes	Spanish	+30 199 576 5038
3	Jandy Ghion	2020-04-08	Hypertension	Spanish	+420 510 814 3677
4	Jason Pitten	2018-07-24	Asthma	Spanish	+234 596 345 8081
5	Jourdain Humphrys	2007-04-26	Diabetes	English	+1 302 811 9389
6	Jim Dimmock	2004-09-30	None	Spanish	+62 408 415 5641
7	Jeddah Hagland	2015-10-08	Asthma	Spanish	+62 829 103 3677
8	Jennine Scotcher	2012-09-25	Hypertension	French	+351 292 108 8579
9	Jodi Sherwill	2011-08-20	Asthma	Spanish	+86 110 433 6216
10	Jayne Pointon	2017-03-19	Hypertension	German	+34 350 789 8516
11	Judie Shinton	2004-02-14	Asthma	French	+62 151 958 8624
12	Jacqui Jenoure	2015-09-08	Hypertension	Spanish	+389 150 421 3014

Total rows: 54 of 54 Query complete 00:00:00.994

Successfully run. Total query runtime: 94 msec. 54 rows affected.

Query returned successfully in 65 msec.

Query returned successfully in 54 msec.

Ln 1, Col 1

```
SELECT * FROM Customers WHERE Name LIKE 'J%';
```

15. Aggregate Functions: Calculate the total cost of all services.

ANS :-

```
1 SELECT SUM(cost) AS total_cost FROM Services;
```

total_cost
71871.54

Total rows: 1 of 1 Query complete 00:00:00.085

Successfully run. Total query runtime: 85 msec. 1 rows affected.

Query returned successfully in 65 msec.

Query returned successfully in 54 msec.

Ln 1, Col 46

SELECT SUM(cost) AS total_cost FROM Services;

16. MAX() and MIN() Functions: Find the most expensive and least expensive services.

ANS :-

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying a tree structure of database objects including Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16). The Tables node is expanded, showing Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, admin, caregivercontacts, caregivers, client, clientcontacts, messages, services, trigger functions, types, and views. The right pane contains a query editor window with the following content:

```
1 SELECT MAX(cost) AS max_cost, MIN(cost) AS min_cost FROM Services;
```

Below the query editor is a Data Output tab showing the results of the query:

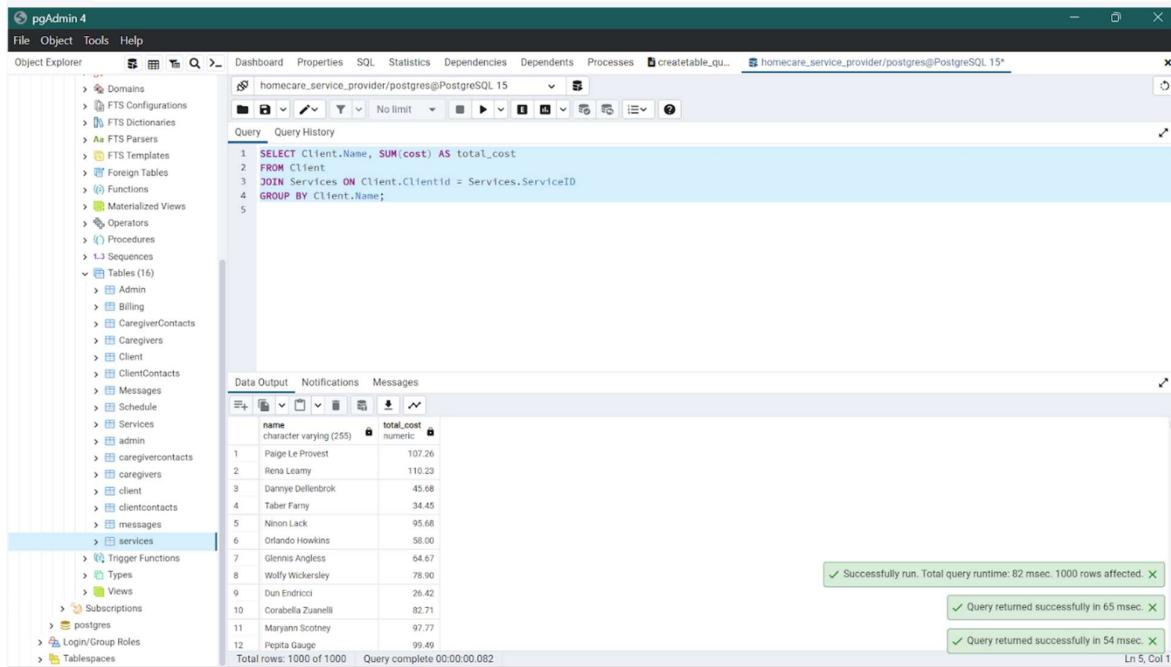
	max_cost	min_cost
1	119.94	20.06

At the bottom of the pgAdmin window, there are three green success messages: "Successfully run. Total query runtime: 60 msec. 1 rows affected.", "Query returned successfully in 65 msec.", and "Query returned successfully in 54 msec.". The status bar at the bottom indicates "Total rows: 1 of 1" and "Query complete 00:00:00.060".

SELECT MAX(cost) AS max_cost, MIN(cost) AS min_cost FROM Services;

17. SUM() Function: Calculate the total cost of services for each customer.

ANS :-



```

SELECT Client.Name, SUM(cost) AS total_cost
FROM Client
JOIN Services ON Client.ClientID = Services.ServiceID
GROUP BY Client.Name;

```

	name	total_cost
1	Paige Le Provest	107.26
2	Rena Leaney	110.23
3	Darnye Dellenbrook	45.68
4	Taber Farny	34.45
5	Ninon Lack	95.68
6	Orlando Hopkins	58.00
7	Glennis Angles	64.67
8	Wally Wickersley	78.90
9	Dun Enducci	26.42
10	Corabella Zuanelli	82.71
11	Maryann Scotney	97.77
12	Peppita Gauge	99.49

Total rows: 1000 of 1000 Query complete 00:00:00.082

✓ Successfully run. Total query runtime: 82 msec. 1000 rows affected.

✓ Query returned successfully in 65 msec.

✓ Query returned successfully in 54 msec.

Ln 5, Col 1

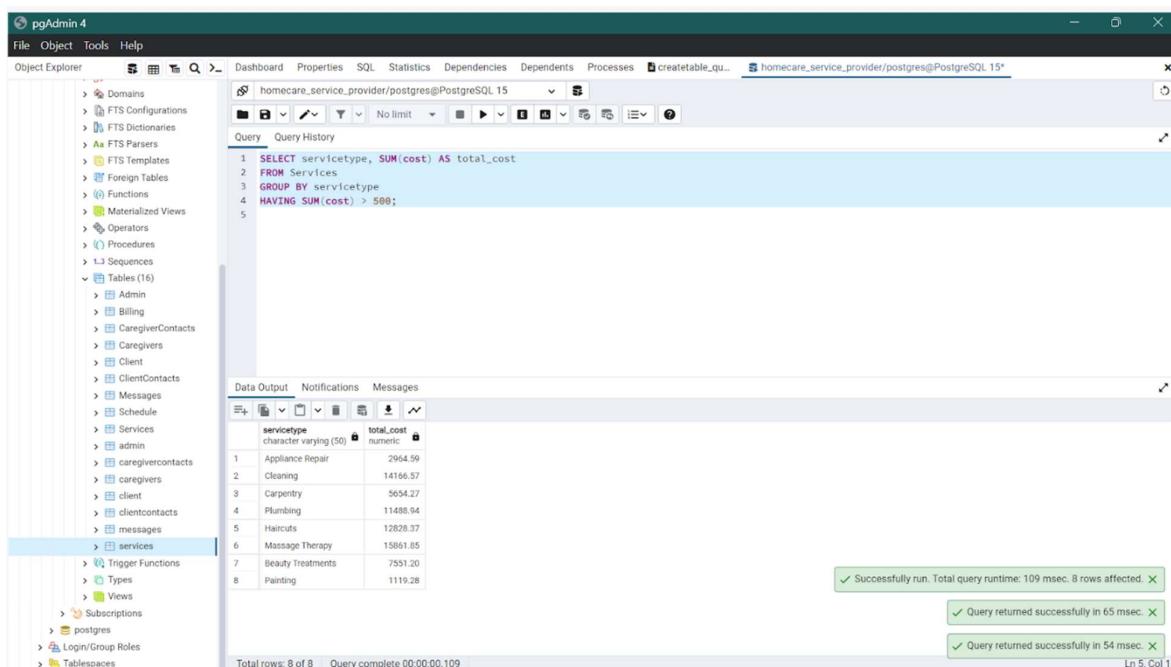
```

SELECT Customers.Name, SUM(cost) AS total_cost
FROM Customers
JOIN Services ON Customers.CustomerID = Services.CustomerID
GROUP BY Customers.Name;

```

18. HAVING Clause: Find service types with a total cost greater than 500.

ANS :-



```

SELECT servicetype, SUM(cost) AS total_cost
FROM Services
GROUP BY servicetype
HAVING SUM(cost) > 500;

```

	servicetype	total_cost
1	Appliance Repair	2964.59
2	Cleaning	14165.57
3	Carpentry	5654.27
4	Plumbing	11488.94
5	Haircut	12828.37
6	Massage Therapy	15861.85
7	Beauty Treatments	7551.20
8	Painting	1119.28

Total rows: 8 of 8 Query complete 00:00:00.109

✓ Successfully run. Total query runtime: 109 msec. 8 rows affected.

✓ Query returned successfully in 65 msec.

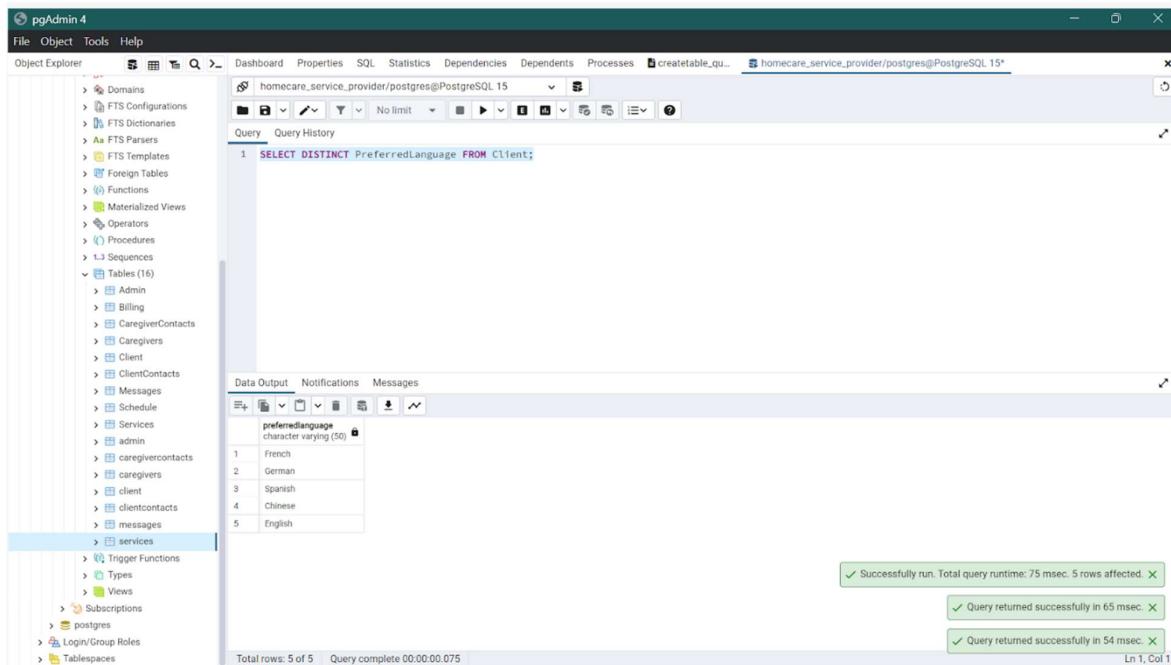
✓ Query returned successfully in 54 msec.

Ln 5, Col 1

```
SELECT servicetype, SUM(cost) AS total_cost
FROM Services
GROUP BY servicetype
HAVING SUM(cost) > 500;
```

19. Distinct Values: List all unique preferred languages among customers.

ANS:-



The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying a tree structure of database objects including Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16). The 'Tables' node is expanded, showing Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, and various contact and message tables. The right pane is the Query Editor, showing the following SQL query:

```
1 SELECT DISTINCT PreferredLanguage FROM Client;
```

The Data Output tab shows the results of the query:

preferredlanguage
French
German
Spanish
Chinese
English

Below the table, there are three green success messages: "Successfully run. Total query runtime: 75 msec. 5 rows affected.", "Query returned successfully in 65 msec.", and "Query returned successfully in 54 msec.". The status bar at the bottom indicates "Total rows: 5 of 5" and "Query complete 00:00:00.075".

```
SELECT DISTINCT PreferredLanguage FROM Customers;
```

20. Self-Join: Find customers who share the same emergency contact.

ANS:-

The screenshot shows the pgAdmin 4 interface. The left sidebar is titled 'Object Explorer' and lists various database objects: Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16). The 'Tables (16)' section is expanded, showing Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, admin, caregivercontacts, caregivers, client, clientcontacts, messages, and services. The main panel contains a 'Query' tab with the following SQL code:

```

1 SELECT c1.Name AS Customer1, c2.Name AS Customer2, c1.EmergencyContacts
2 FROM Client c1
3 JOIN Client c2 ON c1.EmergencyContacts = c2.EmergencyContacts AND c1.ClientID < c2.ClientID;
4

```

The results pane shows three columns: customer1, customer2, and emergencycontacts. Below the results, two green success messages are displayed: 'Query returned successfully in 65 msec.' and 'Query returned successfully in 54 msec.'.

```

SELECT c1.Name AS Customer1, c2.Name AS Customer2, c1.EmergencyContacts
FROM Customers c1
JOIN Customers c2 ON c1.EmergencyContacts = c2.EmergencyContacts AND c1.CustomerID
< c2.CustomerID;

```

21. Union: Combine results of customers who prefer 'English' and those who prefer 'Spanish'.

ANS :-

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows a tree view of database objects including Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16). The 'Tables' node is expanded.
- Query Editor:** Contains the following SQL code:


```

1 SELECT * FROM client WHERE PreferredLanguage = 'English'
2 UNION
3 SELECT * FROM Client WHERE PreferredLanguage = 'Spanish';
        
```
- Data Output:** Displays the results of the query in a tabular format. The columns are:

	clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts
1	721	Lorianne Lipprose	2020-03-19	Diabetes	Spanish	+33 168 180 8531
2	6	Raychel Nutt	2009-01-29	Asthma	Spanish	+86 899 221 6840
3	877	Sula Posnette	2011-04-17	Diabetes	Spanish	+98 412 174 2295
4	124	Suzi Sumeyers	2009-01-13	Hypertension	English	+967 819 359 6629
5	404	Allister Hemp	2013-06-06	Diabetes	Spanish	+86 717 372 5073
6	148	Vonne Tatershall	2019-06-30	Asthma	Spanish	+63 502 369 8678
7	137	Carroll Gullis	2001-08-11	Diabetes	Spanish	+236 980 476 2139
8	215	Stan Mosby	2008-03-23	Diabetes	Spanish	+86 398 609 8062
9	717	Fabien Yukhrin	2000-12-25	Diabetes	Spanish	+351 593 950 2240
10	832	Floyd Wigginton	2016-09-01	Diabetes	Spanish	+54 520 582 8359
11	145	Basilus Poltoone	2007-07-04	Diabetes	Spanish	+7 283 375 0462
12	507	Starr Bedding	2014-10-23	Diabetes	English	+381 601 601 9218

 Total rows: 542 of 542 Query complete 00:00:00.076
- Messages:** Shows three success messages indicating the queries were run successfully.

```

SELECT * FROM Customers WHERE PreferredLanguage = 'English'
UNION
SELECT * FROM Customers WHERE PreferredLanguage = 'Spanish';
        
```

22. Intersection: Find common customers between those who prefer 'English' and 'Spanish'.

ANS :-

```

1 SELECT * FROM client WHERE PreferredLanguage = 'English'
2 INTERSECT
3 SELECT * FROM client WHERE PreferredLanguage = 'Spanish';

```

clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts

Successfully run. Total query runtime: 66 msec. 0 rows affected.

Query returned successfully in 65 msec.

Query returned successfully in 54 msec.

Total rows: 0 of 0 Query complete 00:00:00.066

```

SELECT * FROM Customers WHERE PreferredLanguage = 'English'
INTERSECT
SELECT * FROM Customers WHERE PreferredLanguage = 'Spanish';

```

23. Except: Find customers who prefer 'English' but not 'Spanish'.

ANS :-

```

1 SELECT * FROM client WHERE PreferredLanguage = 'English'
2 EXCEPT
3 SELECT * FROM client WHERE PreferredLanguage = 'Spanish';
4

```

clientid	name	dob	medicalhistory	preferredlanguage	emergencycontacts	
1	140	Yvor Durrett	2011-07-07	Diabetes	English	+54 716 297 0757
2	615	Pete Simon	2020-06-05	No known allergies	English	+57 456 521 0911
3	607	Rena Learny	2021-08-14	Hypertension	English	+55 991 688 6845
4	235	Vasily Spirling	2009-07-17	Diabetes	English	+998 424 154 6756
5	591	Myer Ellaway	2013-04-11	Diabetes	English	+82 216 495 6362
6	700	Chounce Bough	2009-04-18	Hypertension	English	+86 207 950 3181
7	558	Clemmie Kendall	2011-01-09	Hypertension	English	+30 411 864 3045
8	941	Reykah Fern	2019-11-19	None	English	+63 612 502 3071
9	774	Derick Routham	2018-01-30	Diabetes	English	+62 848 310 8588
10	746	Lorallie Deedien	2020-05-22	Diabetes	English	+54 697 172 4334
11	377	Penni Songist	2019-09-17	Diabetes	English	+423 275 166 2196
12	16	Rheba MacIntyre	2004-01-16	Hypertension	English	+7 410 738 8186

Total rows: 219 of 219 Query complete 00:00:00.074

Successfully run. Total query runtime: 74 msec. 219 rows affected.

Query returned successfully in 65 msec.

Query returned successfully in 54 msec.

Ln 4, Col 1

```

SELECT * FROM Customers WHERE PreferredLanguage = 'English'
EXCEPT
SELECT * FROM Customers WHERE PreferredLanguage = 'Spanish';

```

24. Case Statements: Categorize services into 'Affordable' or 'Expensive' based on cost.

ANS :-

The screenshot shows the pgAdmin 4 interface. In the Object Explorer on the left, under the 'Tables (16)' section, the 'services' table is selected. The main area displays a SQL query and its execution results.

```

1 SELECT servicetype, cost,
2 CASE
3     WHEN cost <= 50 THEN 'Affordable'
4     ELSE 'Expensive'
5 END AS cost_category
6 FROM Services;
7

```

The Data Output tab shows the results of the query:

	servicetype	cost	cost_category
1	Plumbing	84.08	Expensive
2	Massage Therapy	40.04	Affordable
3	Massage Therapy	107.95	Expensive
4	Plumbing	68.41	Expensive
5	Massage Therapy	114.91	Expensive
6	Haircuts	119.64	Expensive
7	Plumbing	117.51	Expensive
8	Haircuts	81.88	Expensive
9	Haircuts	76.04	Expensive
10	Beauty Treatments	78.68	Expensive
11	Cleaning	78.03	Expensive
12	Haircuts	61.17	Expensive

Total rows: 1000 of 1000 Query complete 00:00:00.129

Execution status messages:

- Successfully run. Total query runtime: 129 msec. 1000 rows affected.
- Query returned successfully in 65 msec.
- Query returned successfully in 54 msec.

Ln 7, Col 1

```

SELECT servicetype, cost,
CASE
    WHEN cost <= 50 THEN 'Affordable'
    ELSE 'Expensive'
END AS cost_category
FROM Services;

```

25. Window Functions: Rank customers by their total service cost.

ANS :-

```

SELECT Name, SUM(cost) AS total_cost, RANK() OVER (ORDER BY SUM(cost) DESC) AS cost_rank
FROM Client
JOIN Services ON Client.ClientID = Services.ServiceID
GROUP BY Name;

```

	name	total_cost	cost_rank
1	Muffin McCourtie	119.94	1
2	Parry Bellow	119.90	2
3	Benedict Bearon	119.87	3
4	Dorie Yedy	119.83	4
5	Grant Cardell	119.81	5
6	Noel Brumblie	119.77	6
7	Jandy Ghio	119.76	7
8	Raychel Nutt	119.64	8
9	Jesus Fyall	119.59	9
10	Eadie Thomazet	119.58	10
11	Broddy Collisson	119.21	11
12	Suzi Summeyers	119.05	12

Total rows: 1000 of 1000 Query complete 00:00:00.079

Successfully run. Total query runtime: 79 msec. 1000 rows affected.

Query returned successfully in 65 msec.

Query returned successfully in 54 msec.

Ln 5, Col 1

```

SELECT Name, SUM(cost) AS total_cost, RANK() OVER (ORDER BY SUM(cost) DESC)
AS cost_rank
FROM Customers
JOIN Services ON Customers.CustomerID = Services.CustomerID
GROUP BY Name;

```

26. Subquery with Multiple Conditions: Find customers who have used services with a cost greater than the average cost, but less than 200.

ANS :-

```

SELECT * FROM client
WHERE clientID IN (
    SELECT clientID FROM Services
    WHERE cost > (SELECT AVG(cost) FROM Services)
    AND cost < 200
);

```

clientID	name	dob	medicalhistory	preferredlanguage	emergencycontacts
1	Johanna Veldens	2013-04-26	Diabetes	Spanish	+46 567 713 7430
2	Charlot Coen	2008-06-21	Hypertension	Spanish	+81 601 103 0598
3	Perf Tesoe	2003-05-10	Diabetes	English	+34 997 960 4011
4	Sallez Zukerman	2007-01-29	None	Spanish	+86 912 692 5749
5	Timmie Graybeal	2009-06-15	Diabetes	Spanish	+86 999 735 2098
6	Rachel Nut	2009-01-29	Asthma	Spanish	+86 899 221 6840
7	Adil Robun	2016-05-01	Hypertension	English	+1 336 965 6408
8	Kernie Larne	2022-08-22	Asthma	French	+86 539 124 5774
9	Hussein Dunseath	2004-09-11	Diabetes	French	+63 950 665 5910
10	Darleen Brithman	2020-07-03	Diabetes	German	+81 675 566 1871
11	Betteann Hutchinson	2001-12-26	Hypertension	French	+94 994 671 8421
12	Dallis Alvarado	2022-01-08	Diabetes	Spanish	+7 155 878 7161

Total rows: 1000 of 1000 Query complete 00:00:00.083

Successfully run. Total query runtime: 83 msec. 1000 rows affected.

Query returned successfully in 65 msec.

Query returned successfully in 54 msec.

Ln 7, Col 1

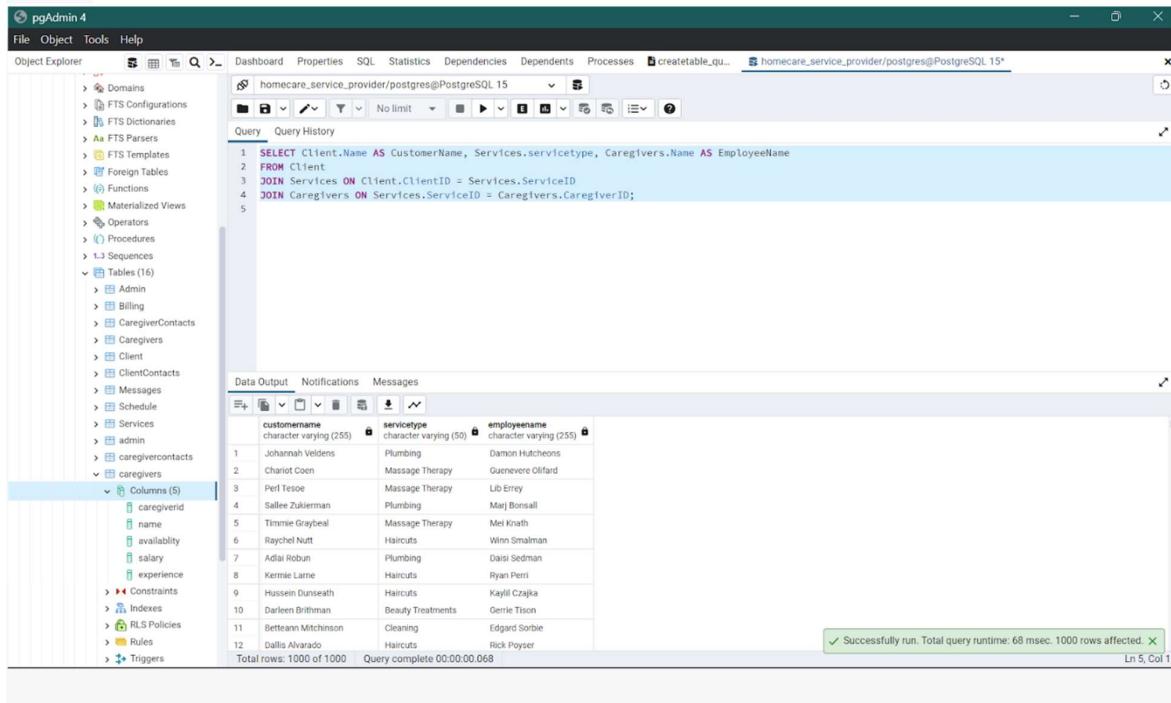
```

SELECT * FROM Customers
WHERE CustomerID IN (
    SELECT CustomerID FROM Services
    WHERE cost > (SELECT AVG(cost) FROM Services)
    AND cost < 200
);

```

27. Joins with Multiple Tables: Retrieve a list of customers, their services, and the employees who provided those services.

ANS :-



```

SELECT Client.Name AS CustomerName, Services.servicetype, Caregivers.Name AS EmployeeName
FROM Client
JOIN Services ON Client.ClientID = Services.ServiceID
JOIN Caregivers ON Services.ServiceID = Caregivers.CaregiverID;

```

The screenshot shows the pgAdmin 4 interface with a query editor window. The query is displayed in the Query tab:

```

SELECT Client.Name AS CustomerName, Services.servicetype, Caregivers.Name AS EmployeeName
FROM Client
JOIN Services ON Client.ClientID = Services.ServiceID
JOIN Caregivers ON Services.ServiceID = Caregivers.CaregiverID;

```

The Data Output tab displays the results of the query:

CustomerName	ServiceType	EmployeeName
Johannah Veldens	Plumbing	Damon Hutcheons
Charlot Coen	Massage Therapy	Guenevere Olfard
Perl Tesoe	Massage Therapy	Lil Errey
Sallee Zukerman	Plumbing	May Bonsall
Timmie Graybeal	Massage Therapy	Mei Knath
Raychel Nutt	Haircuts	Winn Smalman
Adai Robai	Plumbing	Daisi Sedman
Kermie Lame	Haircuts	Ryan Perri
Hussein Dunseath	Haircuts	Kayil Czajka
Darleen Britthman	Beauty Treatments	Gennie Tison
Betteann Mitchinson	Cleaning	Edgard Sorbie
Dallis Alvarado	Haircuts	Rick Poyer

Total rows: 1000 of 1000 Query complete 00:00:00.068

Successfully run. Total query runtime: 68 msec. 1000 rows affected.

```

SELECT Client.Name AS CustomerName, Services.servicetype, Caregivers.Name AS
EmployeeName
FROM Client
JOIN Services ON Client.ClientID = Services.ServiceID
JOIN Caregivers ON Services.ServiceID = Caregivers.CaregiverID;

```

28. Correlated Subquery: Find customers who have used services that are more expensive than the average cost of their preferred service type.

ANS :-

```

1 SELECT *
2 FROM client c
3 WHERE EXISTS (
4   SELECT 1
5   FROM Services s
6   WHERE s.serviceID = c.customerID
7   AND s.cost > (
8     SELECT AVG(cost)
9     FROM Services
10    WHERE servicetype = s.servicetype
11  )
12 );
13

```

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with tables like Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, and various administrative tables.
- Query Editor:** Contains the SQL query provided above.
- Data Output:** Displays the results of the query, which are 507 rows of data from the 'client' table. The columns include clientid, name, dob, medicalhistory, preferredlanguage, and emergencycontacts.
- Messages:** Shows a success message: "Successfully run. Total query runtime: 132 msec. 507 rows affected." and a note: "Ln 13, Col 1".

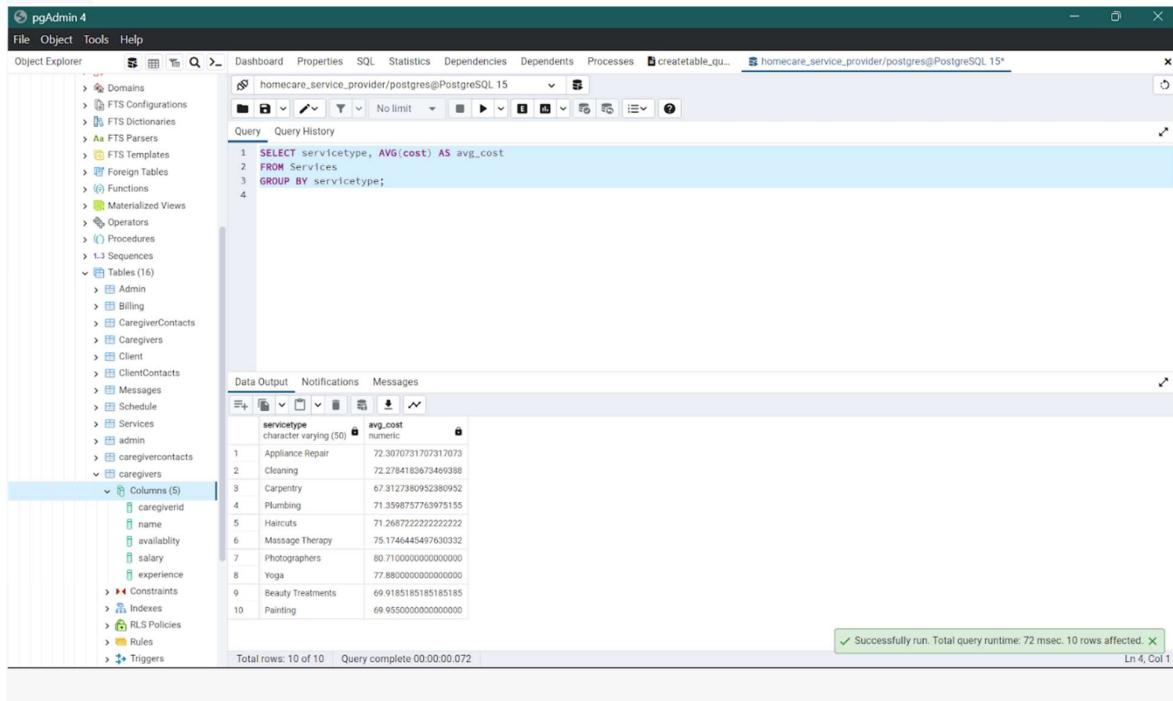
```

SELECT *
FROM Customers c
WHERE EXISTS (
  SELECT 1
  FROM Services s
  WHERE s.CustomerID = c.CustomerID
  AND s.cost > (
    SELECT AVG(cost)
    FROM Services
    WHERE servicetype = s.servicetype
  )
);

```

29. Aggregate Functions with Grouping: Calculate the average cost of services for each service type.

ANS :-



```

1 SELECT servicetype, AVG(cost) AS avg_cost
2 FROM Services
3 GROUP BY servicetype;
4

```

servicetype	avg_cost
Appliance Repair	72.3070731707317073
Cleaning	72.2784185673495938
Carpentry	67.3127380952380952
Plumbing	71.3596757765975155
Haircuts	71.2687222222222222
Massage Therapy	75.1746445497639332
Photographers	80.7100000000000000
Yoga	77.8800000000000000
Beauty Treatments	69.9185185185185185
Painting	69.9350000000000000

Total rows: 10 of 10 Query complete 00:00:00.072 Successfully run. Total query runtime: 72 msec. 10 rows affected. Ln 4, Col 1

```

SELECT servicetype, AVG(cost) AS avg_cost
FROM Services
GROUP BY servicetype;

```

30. Combining Aggregations: Find the service type with the highest total cost and the customer who spent the most on that service type.

ANS :-

The screenshot shows the pgAdmin 4 interface with a database connection named 'homecare_service_provider/postgres@PostgreSQL 15*'. The Object Explorer on the left lists various database objects like domains, FTS configurations, tables, and procedures. The central Query Editor contains the following SQL code:

```

1 WITH ServiceTotalCost AS (
2     SELECT servicetype, SUM(cost) AS total_cost
3     FROM Services
4     GROUP BY servicetype
5 )
6 SELECT s.servicetype, s.total_cost, c.Name AS CustomerName, c.ClientID
7 FROM ServiceTotalCost s
8 JOIN Services se ON s.servicetype = se.servicetype
9 JOIN Client c ON se.CustomerID = c.ClientID
10 WHERE s.total_cost = (SELECT MAX(total_cost) FROM ServiceTotalCost);
11

```

The Data Output tab displays the results of the query, which is a list of service types and their total costs, joined with customer names and IDs. The results are as follows:

	servicetype	total_cost	customername	clientid
1	Massage Therapy	15861.85	Charlot Coen	2
2	Massage Therapy	15861.85	Perf Tesoe	3
3	Massage Therapy	15861.85	Timmie Graybeal	5
4	Massage Therapy	15861.85	Erica Alzirkovich	19
5	Massage Therapy	15861.85	Steffanie Newsome	23
6	Massage Therapy	15861.85	Barbara-anne Neathway	30
7	Massage Therapy	15861.85	Jess Caskey	31
8	Massage Therapy	15861.85	Radora Ghilardini	40
9	Massage Therapy	15861.85	Stoddard Hindhaugh	43
10	Massage Therapy	15861.85	Sindare Gisèle	47
11	Massage Therapy	15861.85	Shana Culmer	49
12	Massage Therapy	15861.85	Wood Pacey	50

Total rows: 211 of 211 Query complete 00:00:00.066

Message bar at the bottom right: Successfully run. Total query runtime: 66 msec. 211 rows affected. Ln 11, Col 1

```

WITH ServiceTotalCost AS (
    SELECT servicetype, SUM(cost) AS total_cost
    FROM Services
    GROUP BY servicetype
)
SELECT s.servicetype, s.total_cost, c.Name AS CustomerName, c.CustomerID
FROM ServiceTotalCost s
JOIN Services se ON s.servicetype = se.servicetype
JOIN Customers c ON se.CustomerID = c.CustomerID
WHERE s.total_cost = (SELECT MAX(total_cost) FROM ServiceTotalCost);

```

31. Using CASE with Aggregation: Calculate the total cost for each service type and categorize them as 'High', 'Medium', or 'Low' cost.

ANS :-

```

SELECT servicetype, SUM(cost) AS total_cost,
CASE
    WHEN SUM(cost) > 500 THEN 'High'
    WHEN SUM(cost) > 250 THEN 'Medium'
    ELSE 'Low'
END AS cost_category
FROM Services
GROUP BY servicetype;

```

	servicetype	total_cost	cost_category
1	Appliance Repair	2964.59	High
2	Cleaning	14166.57	High
3	Carpentry	5654.27	High
4	Plumbing	11488.94	High
5	Haircuts	12828.37	High
6	Massage Therapy	15861.85	High
7	Photographers	80.71	Low
8	Yoga	155.76	Low
9	Beauty Treatments	7551.20	High
10	Painting	1119.28	High

Total rows: 10 of 10 Query complete 00:00:00.145 ✓ Successfully run. Total query runtime: 145 msec. 10 rows affected. Ln 9, Col 1

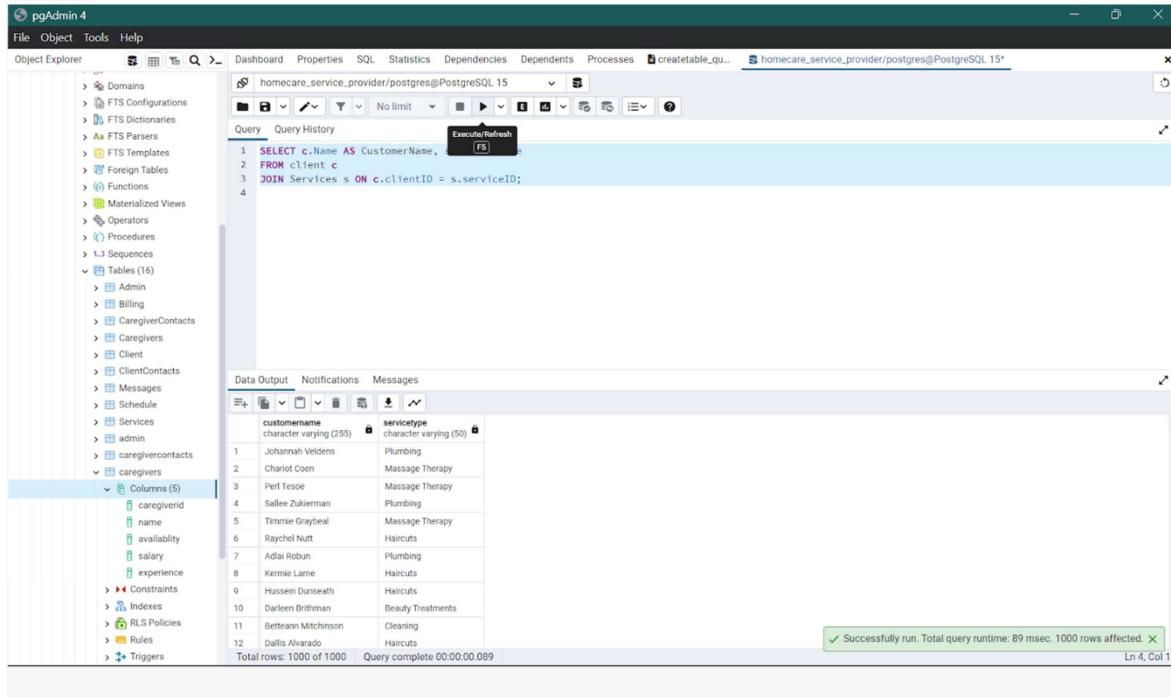
```

SELECT servicetype, SUM(cost) AS total_cost,
CASE
    WHEN SUM(cost) > 500 THEN 'High'
    WHEN SUM(cost) > 250 THEN 'Medium'
    ELSE 'Low'
END AS cost_category
FROM Services
GROUP BY servicetype;

```

32. Joins with Aliases: Retrieve a list of customers and their services, using table aliases.

ANS :-



```

SELECT c.Name AS CustomerName, s.servicetype
FROM Customers c
JOIN Services s ON c.CustomerID = s.CustomerID;

```

	CustomerName	ServiceType
1	Johannah Veldens	Plumbing
2	Charlot Coen	Massage Therapy
3	Perl Tesoe	Massage Therapy
4	Sallee Zukerman	Plumbing
5	Timmie Graybeal	Massage Therapy
6	Raychel Nutt	Haircuts
7	Adai Roba	Plumbing
8	Kermie Lame	Haircuts
9	Hussein Dunseath	Haircuts
10	Darleen Britthman	Beauty Treatments
11	Betteann Mitchinson	Cleaning
12	Dallis Alvarado	Haircuts

Total rows: 1000 of 1000 Query complete 00:00:00.089

Successfully run. Total query runtime: 89 msec. 1000 rows affected.

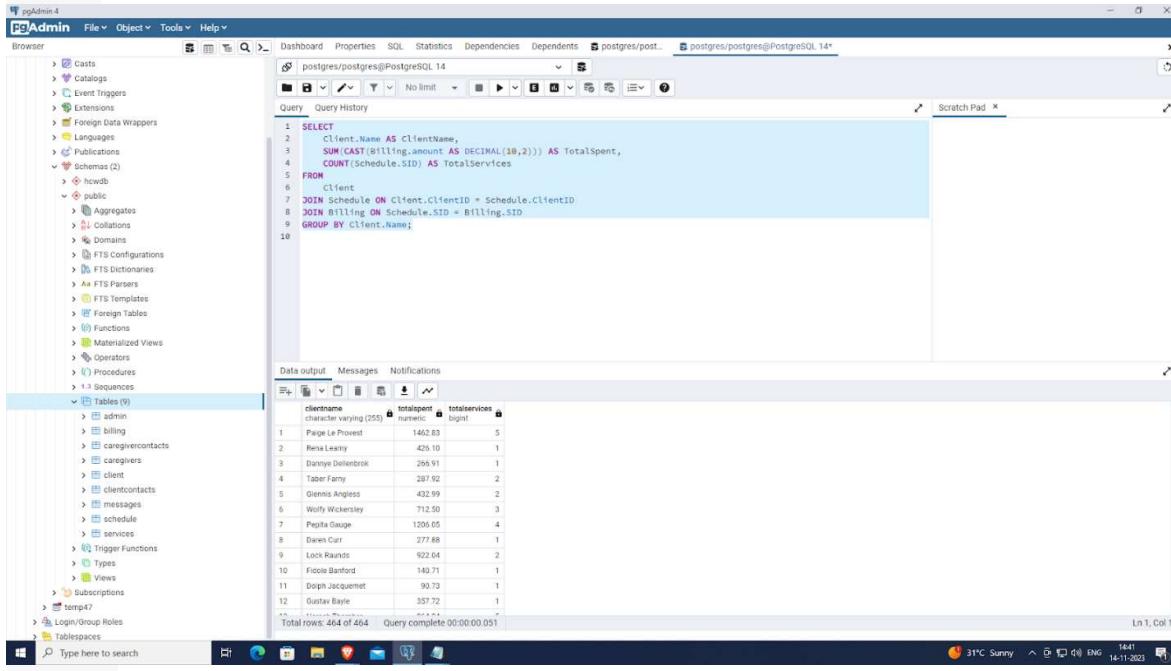
```

SELECT c.Name AS CustomerName, s.servicetype
FROM Customers c
JOIN Services s ON c.CustomerID = s.CustomerID;

```

33. The total amount spent by each client and the number of services they have availed.

ANS :-



```

SELECT
    Client_Name AS ClientName,
    SUM(CAST(Billing.Amount AS DECIMAL(18,2))) AS TotalSpent,
    COUNT(Schedule.SID) AS TotalServices
FROM
    Client
    JOIN Schedule ON Client.ClientID = Schedule.ClientID
    JOIN Billing ON Schedule.SID = Billing.SID
    GROUP BY Client.ClientName;

```

ClientName	TotalSpent	TotalServices
Paige Le Provost	1462.83	5
Reisa Lenity	426.10	1
Danyne Delenbrook	266.91	1
Tabor Farny	287.92	2
Glenis Angless	432.99	2
Wolff Wickerley	712.50	3
Pepita Gaupe	1206.05	4
Daren Curr	277.88	1
Lois Raunis	922.04	2
Fipole Banford	149.71	1
Dolph Jacquemet	90.73	1
Gustav Bayle	357.72	1

Total rows: 464 of 464 Query complete 00:00:00.051

SELECT

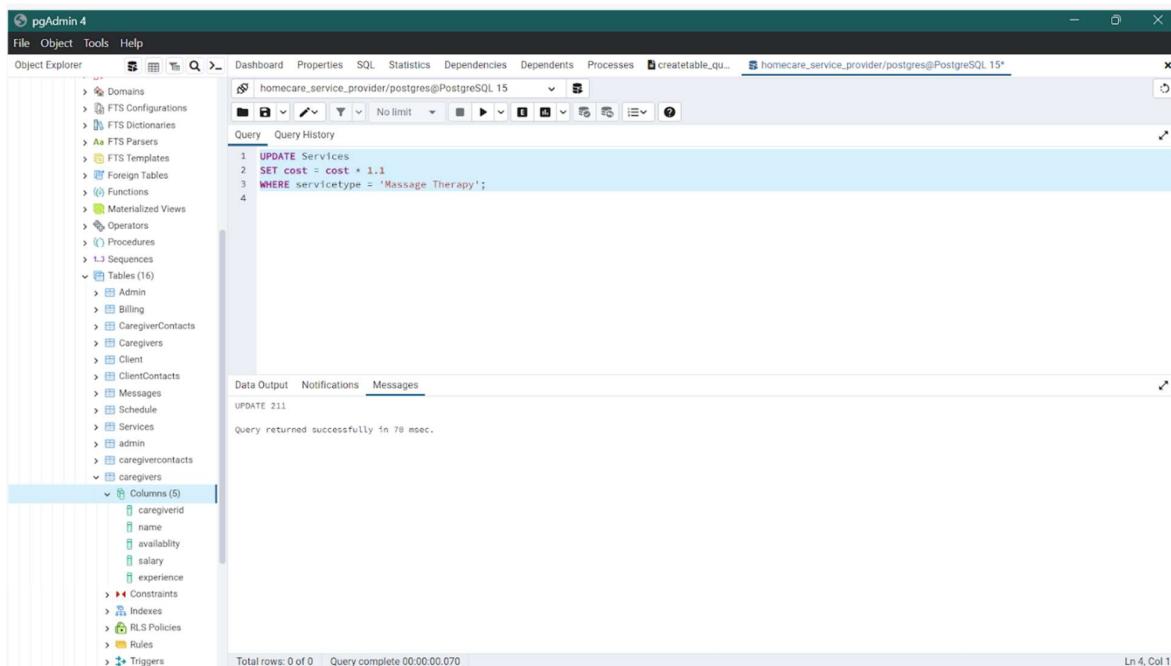
```

Client.Name AS ClientName,
SUM(CAST(Billing.amount AS DECIMAL(10,2))) AS TotalSpent,
COUNT(Schedule.SID) AS TotalServices
FROM
Client
JOIN Schedule ON Client.ClientID = Schedule.ClientID
JOIN Billing ON Schedule.SID = Billing.SID
GROUP BY Client.Name;

```

34. Update Query: Increase the cost of all 'Massage Therapy' services by 10%.

ANS :-



The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like Domains, FTS Configurations, Functions, and Tables (16). The main pane is a query editor window titled 'homecare_service_provider/postgres@PostgreSQL 15*'. It contains the following SQL code:

```

1 UPDATE Services
2 SET cost = cost * 1.1
3 WHERE servicetype = 'Massage Therapy';
4

```

Below the code, the 'Data Output' tab shows the result: 'UPDATE 211'. A message at the bottom states 'Query returned successfully in 78 msec.'

```

UPDATE Services
SET cost = cost * 1.1
WHERE servicetype = 'Massage Therapy';

```

35. Delete Query: Delete all customers who have 'None' as their medical history.

ANS :-

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with various database objects like Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16). The 'Tables' node is expanded, showing Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, and admin, caregivercontacts, caregivers. The 'caregivers' node is selected, and its 'Columns (5)' are listed: caregiverid, name, availability, salary, and experience. The main pane shows a SQL query window with the following code:

```

1 DELETE FROM client
2 WHERE MedicalHistory = 'None';
3

```

The 'Data Output' tab shows the result of the query: 'DELETE 128'. Below it, a message says 'Query returned successfully in 63 msec.'.

**DELETE FROM Customers
WHERE MedicalHistory = 'None';**

36. Triggers: Create a trigger to log changes to the Services table.

ANS :-

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with various database objects. The 'Tables' node is expanded, showing Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, and admin, caregivercontacts, caregivers. The 'caregivers' node is selected, and its 'Columns (5)' are listed. The main pane shows a SQL query window with the following code:

```

1 CREATE OR REPLACE FUNCTION log_service_changes()
2 RETURNS TRIGGER AS $$
3 BEGIN
4   INSERT INTO Servicelog (ServiceID, OldCost, NewCost, ChangeTimestamp)
5   VALUES (OLD.ServiceID, OLD.cost, NEW.cost, NOW());
6   RETURN NEW;
7 END;
8 $$ LANGUAGE plpgsql;
9
10 CREATE TRIGGER service_changes_trigger
11 AFTER UPDATE ON Services
12 FOR EACH ROW
13 EXECUTE FUNCTION log_service_changes();
14

```

The 'Data Output' tab shows the result of the query: 'CREATE TRIGGER'. Below it, a message says 'Query returned successfully in 59 msec.'.

-- Assuming you have a log table named ServiceLog with appropriate columns

```

CREATE OR REPLACE FUNCTION log_service_changes()
RETURNS TRIGGER AS $$

BEGIN
    INSERT INTO ServiceLog (ServiceID, OldCost, NewCost, ChangeTimestamp)
    VALUES (OLD.ServiceID, OLD.cost, NEW.cost, NOW());
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER service_changes_trigger
AFTER UPDATE ON Services
FOR EACH ROW
EXECUTE FUNCTION log_service_changes();

```

37. Stored Procedure: Create a stored procedure to calculate the total cost of services for a specific customer.

ANS :-

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with various database objects like Domains, FTS Configurations, Functions, Procedures, and Tables. The Tables section is expanded, showing tables such as Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, and caregivercontacts. The caregiver table is currently selected, and its columns (caregiverid, name, availability, salary, experience) are listed in the details pane. The main query editor window contains the SQL code for creating the stored procedure:

```

CREATE OR REPLACE FUNCTION calculate_total_cost(customer_id INT)
RETURNS NUMERIC AS $$
DECLARE
    total_cost NUMERIC;
BEGIN
    SELECT SUM(cost) INTO total_cost
    FROM Services
    WHERE CustomerID = customer_id;
    RETURN total_cost;
END;
$$ LANGUAGE plpgsql;

```

The status bar at the bottom indicates "Query returned successfully in 54 msec." and "Ln 12, Col 1".

-- Assuming you have a parameterized stored procedure

```

CREATE OR REPLACE FUNCTION calculate_total_cost(customer_id INT)
RETURNS NUMERIC AS $$

DECLARE
    total_cost NUMERIC;
BEGIN
    SELECT SUM(cost) INTO total_cost
    FROM Services

```

```

WHERE CustomerID = customer_id;
    RETURN total_cost;
END;
$$ LANGUAGE plpgsql;

```

40 .Advanced Joins: Retrieve a list of customers and the services they have used, along with the employees who provided those services, and sort them by the service cost.

ANS :-

customername	servicetype	cost	employeeName
Nicky Winwood	Plumbing	20.06	Julie Zorro
Teresina Yerill	Haircuts	20.10	Gibb Moules
Myra Cicci	Beauty Treatments	20.14	Norina Orto
Marie-jeanne Fairhead	Plumbing	20.21	Philia Hearnson
Katerine Twigg	Haircuts	20.25	Violante Tezure
Kelli Greenside	Haircuts	20.33	Jakob Pickvance
Anna Malinson	Cleaning	20.33	Erhard De Dantse
Israel Ciepluch	Plumbing	20.39	Tomasina Baldelli
Madelene Denisyev	Cleaning	20.39	Dottie Peete
Gabi Windrus	Cleaning	20.74	Val Blackwood
Rick Sketch	Plumbing	20.74	Knox Glasner
Kory Japp	Cleaning	21.03	Rozanna Jirneck

Successfully run. Total query runtime: 90 msec. 872 rows affected. Ln 6, Col 1

```

SELECT c.Name AS CustomerName, s.servicetype, s.cost, e.Name AS EmployeeName
FROM client c
JOIN Services s ON c.clientID = s.serviceID
JOIN caregivers e ON s.serviceID = e.caregiverID
ORDER BY s.cost;

```

41. Each caregiver's workload in terms of the number of jobs assigned and how many of those jobs have been paid for.

ANS :-

The screenshot shows the pgAdmin 4 interface. In the top navigation bar, 'File', 'Object', 'Tools', and 'Help' are visible. The main window has a 'Browser' pane on the left containing various database objects like 'Tables', 'Functions', and 'Sequences'. The central area is the 'Query History' tab, which displays the following SQL query:

```

1 SELECT
2     Caregivers.Name AS CaregiverName,
3     COUNT(Schedule.SID) AS JobsAssigned,
4     SUM(CASE WHEN Billing.PaymentStatus = 'Paid' THEN 1 ELSE 0 END) AS JobsPaid
5 FROM
6     Caregivers
7 JOIN Schedule ON Caregivers.CaregiverID = Schedule.CaregiverID
8 JOIN Billing ON Schedule.SID = Billing.SID
9 GROUP BY Caregivers.Name;
10

```

The 'Data output' tab shows the results of the query:

CaregiverName	jobsassigned	jobspaid
Jeremy Finney	1	1
Goeff Emma	2	0
Fada Corcoran	2	2
Sig Yesmin	1	1
Rosaline Kenn	1	0
Gene Whittington	1	1
Shell Cartledge	3	2
Fergus Fowle	2	1
Kath Beddoes	1	0
Christopher Coleson	1	0
Sherne Uccello	1	0
Wes Verralls	6	0

Total rows: 461 of 461 Query complete 00:00:00.093 Successfully run. Total query runtime: 51 msec. 461 rows affected.

```

SELECT
    Caregivers.Name AS CaregiverName,
    COUNT(Schedule.SID) AS JobsAssigned,
    SUM(CASE WHEN Billing.PaymentStatus = 'Paid' THEN 1 ELSE 0 END) AS JobsPaid
FROM
    Caregivers
JOIN Schedule ON Caregivers.CaregiverID = Schedule.CaregiverID
JOIN Billing ON Schedule.SID = Billing.SID
GROUP BY Caregivers.Name;

```

42. Calculate Employee Average Service Cost: Calculate the average cost of services provided by each employee.

ANS:-

```

SELECT e.Name AS EmployeeName, AVG(s.cost) AS AvgServiceCost
FROM Employees e
LEFT JOIN Services s ON e.EmployeeID = s.EmployeeID
GROUP BY e.Name;

```

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with tables like Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, and admin.
- Query Editor:** Contains the SQL query above.
- Data Output:** Displays the results of the query, showing 12 rows of data where each row represents an employee and their average service cost.
- Message Bar:** Shows "Successfully run. Total query runtime: 98 msec. 1000 rows affected."

	EmployeeName	AvgServiceCost
1	Enrica Olivello	121.25000000000000
2	Jeramey Finney	52.61000000000000
3	Talya Matisoff	63.43000000000000
4	Joscelin Mangenot	21.62000000000000
5	Geoff Emms	39.28000000000000
6	Eada Colcomb	69.89000000000000
7	Hendrika Reading	43.96000000000000
8	Sig Yesinia	37.64000000000000
9	Emilie Mattes	41.69000000000000
10	Rosalinde Kensi	100.15000000000000
11	Gene Whittington	85.71000000000000
12	Shell Cuttridge	27.09000000000000

```

SELECT e.Name AS EmployeeName, AVG(s.cost) AS AvgServiceCost
FROM Employees e
LEFT JOIN Services s ON e.EmployeeID = s.EmployeeID
GROUP BY e.Name;

```

43. Identifies unpaid invoices and provides details of the involved clients and caregivers.

ANS :-

```

SELECT
    Billing.InvoiceID,
    Billing.Amount,
    Client.Name AS ClientName,
    Caregiver.Name AS CaregiverName
FROM
    Billing
JOIN Schedule ON Billing.SID = Schedule.SID
JOIN Client ON Schedule.ClientID = Client.ClientID
JOIN Caregiver ON Schedule.CaregiverID = Caregiver.CaregiverID
WHERE
    Billing.PaymentStatus = 'Unpaid';

```

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with tables like admin, billing, caregivercontacts, caregivers, client, clientcontacts, messages, schedule, services, and temp47.
- Query Editor:** Contains the SQL query above.
- Data Output:** Displays the results of the query, showing 12 rows of data where each row represents an unpaid invoice and its associated client and caregiver.

	InvoiceID	Amount	ClientName	CaregiverName
1	3	534.29	Haze Veronique	Kacey Hemmish
2	5	210.79	Wood Pacey	Bobbie Balach
3	7	360.79	Parisal Ilyson	Tybie Oasi
4	8	353.44	Kurt Haney'	Pyrth Lineman
5	10	147.97	Dominic Bebrick	Marsilia Faules
6	12	174.52	Edouard Roskam	Belle Brooks
7	13	421.11	Valerie Clayworth	Michella Dutch
8	15	533.31	Dallas Warner	Gale Quaf
9	17	290.44	Lloyd Sworne	Worthington O'Nolan
10	18	490.88	Ginny Gadditi	Gerty Wynd
11	19	343.71	Gilbert Lupulin	Maisley Hode
12	20	196.71	Gertrude Cossin	Yard Salazar

```

SELECT
    Billing.InvoiceID,
    Billing.Amount,
    Client.Name AS ClientName,
    Caregivers.Name AS CaregiverName
FROM
    Billing
JOIN Schedule ON Billing.SID = Schedule.SID
JOIN Client ON Schedule.ClientID = Client.ClientID
JOIN Caregivers ON Schedule.CaregiverID = Caregivers.CaregiverID
WHERE
    Billing.PaymentStatus = 'Unpaid';

```

44. Services Provided by Employee: List services provided by each employee.

ANS :-

```

SELECT e.Name AS EmployeeName, ARRAY_AGG(s.servicetype) AS ProvidedServices
FROM caregivers e
LEFT JOIN Services s ON e.caregiverID = s.serviceID
GROUP BY e.Name;

```

EmployeeName	ProvidedServices
Enrika Olivello	{Massage Therapy'}
Jeramey Finney	{Plumbing}
Talya Matissoff	{Harcuts}
Joscelin Mangenot	{Plumbing}
Goeff Emma	{Plumbing}
Eada Colcomb	{Carpentry}
Hendrika Reading	{Cleaning}
Sig Yesinia	{Massage Therapy'}
Emilie Mattes	{Massage Therapy'}
Rosalinde Kent	{Appliance Repair'}
Gene Whittington	{Cleaning}
Shell Cuttridge	{Beauty Treatments'}

Total rows: 1000 of 1000 Query complete 00:00:00.084

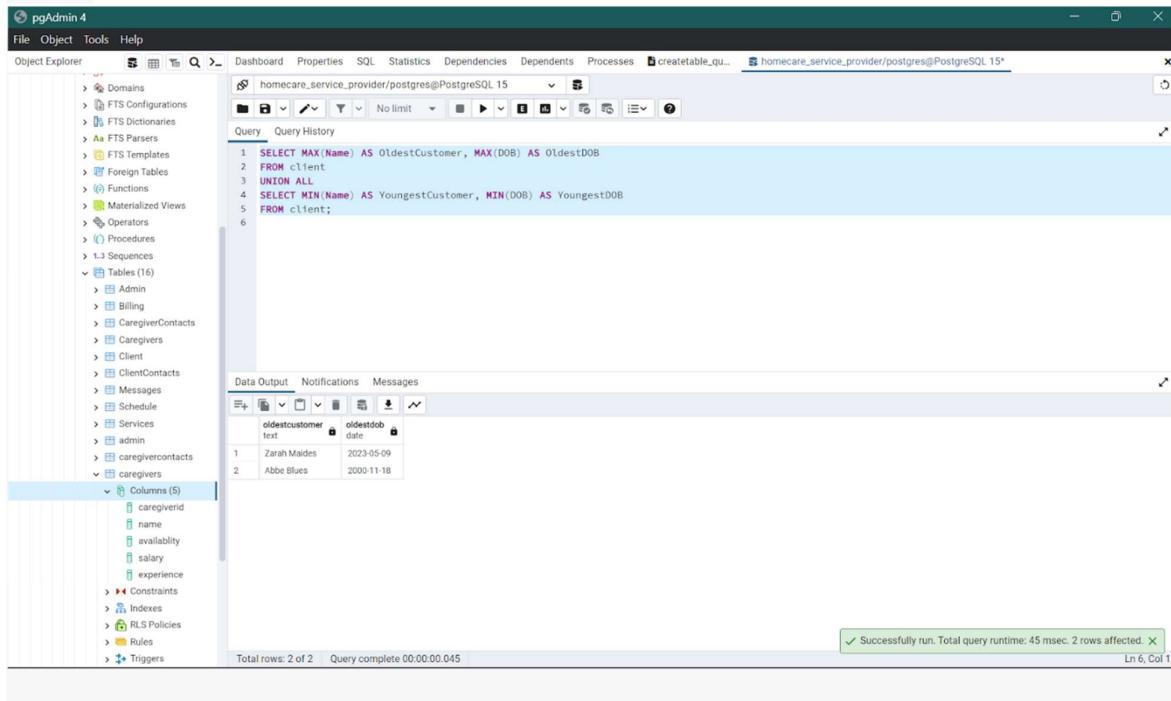
```

SELECT e.Name AS EmployeeName, ARRAY_AGG(s.servicetype) AS ProvidedServices
FROM Employees e
LEFT JOIN Services s ON e.EmployeeID = s.EmployeeID
GROUP BY e.Name;

```

45. Find the Oldest and Youngest Customers: Identify the oldest and youngest customers.

ANS :-



```

1 SELECT MAX(Name) AS OldestCustomer, MAX(DOB) AS OldestDOB
2 FROM client
3 UNION ALL
4 SELECT MIN(Name) AS YoungestCustomer, MIN(DOB) AS YoungestDOB
5 FROM client;
6

```

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like Domains, FTS Configurations, FTS Dictionaries, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16). The Tables section is expanded, showing Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, and caregivercontacts. The caregivercontacts table is selected, and its columns (caregiverid, name, availability, salary, experience) are listed. The main pane shows a query editor with the above SQL code. Below the code is a Data Output grid with two rows:

	oldestcustomer	oldestdob
1	Zarah Maides	2023-05-09
2	Abbe Blues	2000-11-18

At the bottom right of the pgAdmin window, there is a message: "Successfully run. Total query runtime: 45 msec. 2 rows affected. Ln 6, Col 1".

```

SELECT MAX(Name) AS OldestCustomer, MAX(DOB) AS OldestDOB
FROM Customers
UNION ALL
SELECT MIN(Name) AS YoungestCustomer, MIN(DOB) AS YoungestDOB
FROM Customers;

```

46. Provides contact details for each client, including phone, email, and address.

ANS :-

```

SELECT
  Client.Name AS ClientName,
  ClientContacts.phone,
  ClientContacts.email,
  ClientContacts.address
FROM
  Client
JOIN ClientContacts ON Client.ClientID = ClientContacts.ClientID;

```

	clientname	phone	email	address
1	Phoenicia Rivelli	+20 218 631 8860	lrogen@ngov.net	3 Grim Pass
2	Audi Manger	+232 328 980 7396	gyell@ngberkeley.edu	639 Riedy Hill
3	Georgi Hartigan	+86 122 580 8420	cfennelow@multiplay.c...	2 Service Plaza
4	Munroe Rowberry	+62 558 157 9397	ysontor@washington...	2396 Chinno Road
5	Cassandra Neesegen	+7 965 513 4393	nxelco@zyouu.com	6849 Linden Court
6	Anilese Benocchi	+81 180 490 0346	cbradbrook3@123-reg...	46 Oxbridge Pass
7	Austin Matlys	+381 400 179 1911	jlemaud4@apexpost.jp	30 Waxwing Park
8	Aura Orkney	+62 569 672 3758	sdayer@qwest.edu	45937 Tomscot Trail
9	Coralyn Crofts	+63 111 786 3334	pmasereek@slashdot.o...	9 Quincy Street
10	Benoite Marchent	+62 653 801 6912	mwendan7@about.me	64059 Sherman Juncti...
11	Ainslee Brarkey	+55 706 521 7730	tsharkey@yolasite.com	17 Carey Court
12	Nardi Littimore	+55 758 634 4331	noulogh@pcworld.com	864 Burning Wood Plaza

Total rows: 1000 of 1000 Query complete 00:00:00.043 Ln 9, Col 1

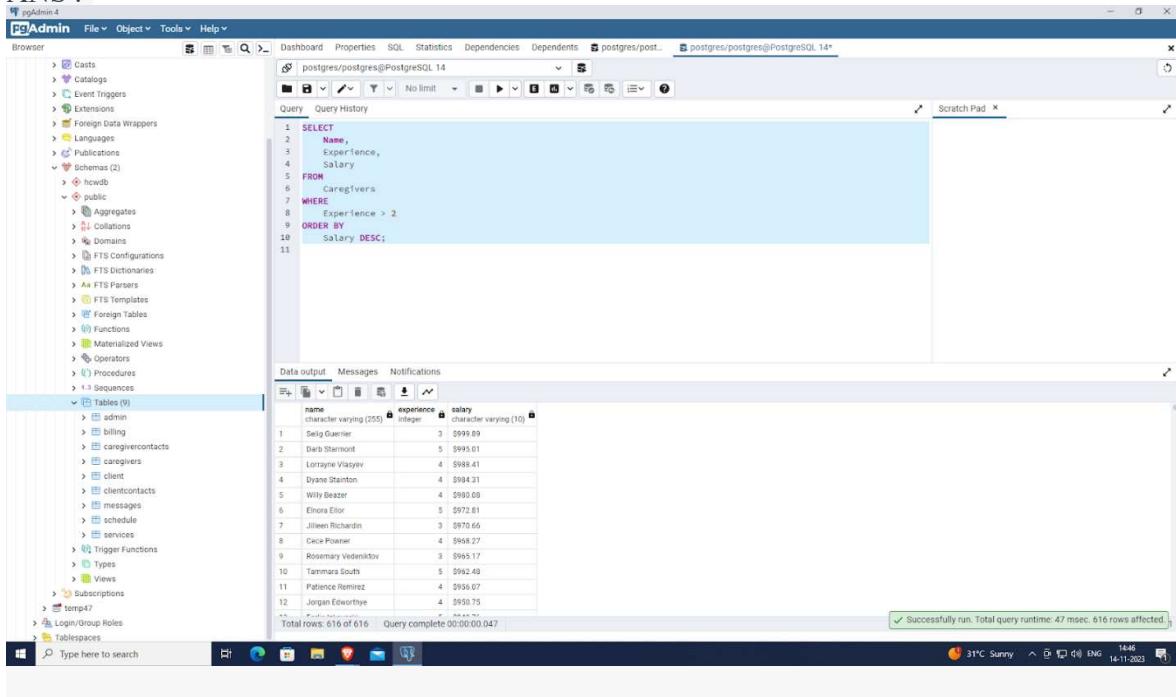
```

SELECT
  Client.Name AS ClientName,
  ClientContacts.phone,
  ClientContacts.email,
  ClientContacts.address
FROM
  Client
JOIN ClientContacts ON Client.ClientID = ClientContacts.ClientID;

```

47. Lists caregivers with more than 2 years of experience, sorted by their salary in descending order.

ANS :-



```

SELECT
  Name,
  Experience,
  Salary
FROM
  Caregivers
WHERE
  Experience > 2
ORDER BY
  Salary DESC;

```

	name	experience	salary
1	Seig Guenier	3	\$993.89
2	Darla Sternmont	5	\$995.01
3	Lorraine Visayev	4	\$988.41
4	Dyane Stanton	4	\$984.31
5	Willy Beazer	4	\$980.08
6	Elvira Dilar	5	\$972.81
7	Jilene Richardin	3	\$970.66
8	Cecia Pownel	4	\$968.27
9	Rosemary Vedenikov	3	\$963.17
10	Tammara South	5	\$962.49
11	Patience Ramirez	4	\$956.07
12	Jorgan Eworthye	4	\$950.75

Total rows: 616 of 616 Query complete: 00:00:00.047 Successfully run. Total query runtime: 47 msec. 616 rows affected.

```

SELECT
  Name,
  Experience,
  Salary
FROM
  Caregivers
WHERE
  Experience > 2
ORDER BY
  Salary DESC;

```

48. Calculate Age from Date of Birth: Calculate the age of each customer based on their date of birth.

ANS:-

```

SELECT Name AS CustomerName, DOB,
       DATE_PART('year', CURRENT_DATE) - DATE_PART('year', DOB) AS Age
FROM Client;

```

	customername	dob	age
1	Johannah Veldens	2013-04-26	10
2	Charlot Coen	2009-06-21	15
3	Perl Tesoe	2003-05-10	20
4	Timmie Graybeal	2009-06-15	14
5	Raychel Nutt	2009-01-29	14
6	Adrai Robun	2016-05-01	7
7	Kermie Lame	2022-08-22	1
8	Hussein Dunseath	2004-09-11	19
9	Darleen Britzman	2020-07-03	3
10	Betteann Mitchinson	2001-12-26	22
11	Dallis Alvarado	2022-01-08	1
12	Carson Rowbrey	2020-06-10	3

Successfully run. Total query runtime: 62 msec. 872 rows affected. Ln 4, Col 1

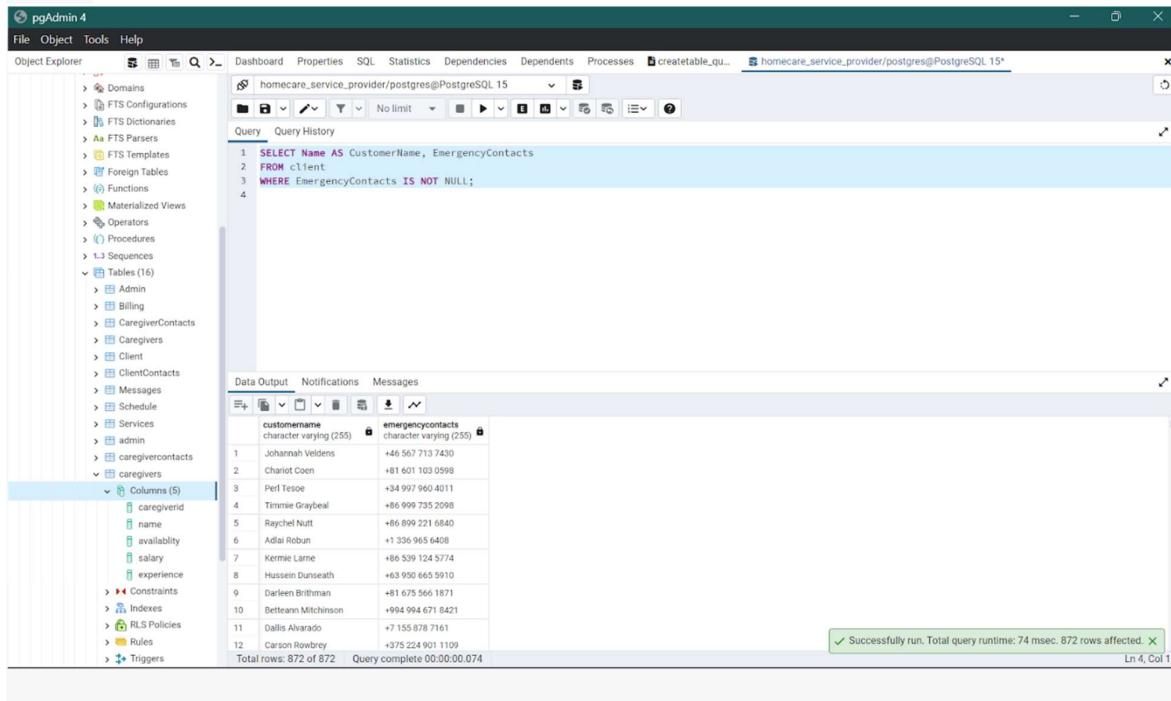
```

SELECT Name AS CustomerName, DOB,
       DATE_PART('year', CURRENT_DATE) - DATE_PART('year', DOB) AS Age
FROM Customers;

```

49. Find Customers with Emergency Contacts: Retrieve customers who have specified emergency contacts.

ANS :-



```

SELECT Name AS CustomerName, EmergencyContacts
FROM Client
WHERE EmergencyContacts IS NOT NULL;

```

	CustomerName	EmergencyContacts
1	Johannah Veldens	+46 567 713 7430
2	Charlot Coen	+81 601 103 0598
3	Perl Tesoe	+34 997 960 4011
4	Timmie Graybeal	+86 999 735 2098
5	Raychel Nutt	+86 899 221 6840
6	Adrai Robun	+1 336 965 6408
7	Kermie Lame	+86 539 124 5774
8	Hussein Dunseath	+63 950 665 5910
9	Darleen Britzman	+81 675 566 1871
10	Betteann Mitchinson	+994 994 671 8421
11	Dallis Alvarado	+71 155 878 7161
12	Carson Rowbrey	+375 224 901 1109

Total rows: 872 of 872 Query complete 00:00:00.074

Successfully run. Total query runtime: 74 msec. 872 rows affected. Ln 4, Col 1

SELECT Name AS CustomerName, EmergencyContacts
 FROM Customers
 WHERE EmergencyContacts IS NOT NULL;

50. Top N Most Expensive Services: Retrieve the top 5 most expensive services.

ANS:-

```

SELECT servicetype, cost
FROM Services
ORDER BY cost DESC
LIMIT 5;

```

	servicetype	cost
1	Massage Therapy	131.89
2	Massage Therapy	131.79
3	Massage Therapy	131.75
4	Massage Therapy	131.74
5	Massage Therapy	131.13

Total rows: 5 of 5 Query complete 00:00:00.093 Successfully run. Total query runtime: 93 msec. 5 rows affected. Ln 5, Col 1

```

SELECT servicetype, cost
FROM Services
ORDER BY cost DESC
LIMIT 5;

```

51. Find Services That Were Never Used: Identify service types that have never been used by any customer.

ANS :-

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16). The 'Tables' node is expanded, showing Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, and three temporary tables: admin, caregivercontacts, and caregivers. The 'caregivers' table is currently selected, and its columns (caregiverid, name, availability, salary, experience) are listed under 'Columns (5)'. The main pane is the Query Editor, which contains the following SQL code:

```
1 SELECT servicetype
2 FROM Services
3 WHERE ServiceID IS NULL;
```

The Data Output tab shows the results of the query:

servicetype
character varying (50)

Below the Data Output tab, the status bar indicates: Total rows: 0 of 0 | Query complete 00:00:00.101 | Successfully run. Total query runtime: 101 msec. 0 rows affected. | Ln 4, Col 1.

```
SELECT servicetype
FROM Services
WHERE ServiceID IS NULL;
```

52. How many clients have utilized each service type.

ANS :-

The screenshot shows the pgAdmin 4 interface. In the left sidebar, the 'Tables' section is expanded, showing various tables like admin, billing, caregivercontacts, caregivers, client, clientcontacts, messages, schedule, services, types, and views. The 'Services' table is selected. In the main area, a SQL query is written in the Query Editor:

```

1 SELECT
2   Services.ServiceType,
3   COUNT(Schedule.ClientID) AS NumberOfClients
4   FROM
5     Services
6   JOIN Schedule ON Services.ServiceID = Schedule.ServiceID
7   GROUP BY Services.ServiceType;
8

```

The Data output tab displays the results of the query:

ServiceType	NumberOfClients
Appliance Repair	44
Cleaning	298
Carpentry	94
Plumbing	142
Haircuts	199
Massage Therapy	198
Yoga	1
Beauty Treatments	100
Painting	14

Total rows: 9 of 9 Query complete 00:00:00.084 Successfully run. Total query runtime: 84 msec. 9 rows affected.

```

SELECT
  Services.ServiceType,
  COUNT(Schedule.ServiceID) AS ServiceCount
FROM
  Services
JOIN Schedule ON Services.ServiceID = Schedule.ServiceID
GROUP BY Services.ServiceType
ORDER BY ServiceCount DESC;

```

53. Identifies the most active caregiver in terms of communication, which could indicate engagement or client issues.

ANS :-

The screenshot shows the pgAdmin 4 interface. In the top navigation bar, 'File', 'Object', 'Tools', and 'Help' are visible. The main window has a 'Browser' tab on the left containing a tree view of database objects like 'Tables', 'Views', 'Functions', etc. The central area is divided into 'Query History' and 'Data output'. The 'Query History' pane contains the following SQL code:

```

1 SELECT
2     Client.Name,
3     COUNT(Messages.messageID) AS TotalMessages
4 FROM
5     Client
6 JOIN Messages ON Client.ClientID = Messages.caregiverid
7 GROUP BY Client.ClientID
8 ORDER BY TotalMessages DESC;
9 select * from messages;

```

The 'Data output' pane displays the results of the query, which is a table with two columns: 'name' and 'totalmessages'. The data is as follows:

	name	totalmessages
1	Andréthe Lombard	6
2	Base Bookmaster	5
3	Cornell Fitzpayn	5
4	Bér Levet	4
5	Shanree Metrick	4
6	Fred Scipps	4
7	Lavinia Trill	4
8	Tatheus Lotthead	4
9	Burt Guppy	4
10	Bink Riosid	4
11	Wainwright McIoure	4
12	Roxine Hamshire	4

Total rows: 634 of 634 Query complete 00:00:00.054

```

SELECT
    Client.Name,
    COUNT(Messages.messageID) AS TotalMessages
FROM
    Client
JOIN Messages ON Client.ClientID = messages.caregiverid
GROUP BY Client.ClientID
ORDER BY TotalMessages DESC;

```

54. Calculate the Average Age of Customers: Calculate the average age of all customers

ANS :-

```

1 SELECT AVG(DATE_PART('year', CURRENT_DATE) - DATE_PART('year', DOB)) AS AvgAge
2 FROM Client;
3

```

Total rows: 1 of 1 Query complete 00:00:00.076 Successfully run. Total query runtime: 76 msec. 1 rows affected.

```

SELECT AVG(DATE_PART('year', CURRENT_DATE) - DATE_PART('year', DOB)) AS
AvgAge
FROM Customers;

```

55. An overview of caregiver availability, which is crucial for scheduling and resource allocation.

ANS:-

The screenshot shows the pgAdmin 4 interface. In the left sidebar, the 'Tables' section is selected, showing various tables like admin, billing, caregivercontacts, caregivers, client, clientcontacts, messages, schedule, services, Trigger Functions, Types, Views, Subscriptions, temp47, and Tablespaces. The 'Caregivers' table is expanded. In the main area, a SQL query is written in the Query Editor:

```

SELECT
    Availability,
    COUNT(*) AS NumberOfCaregivers
FROM
    Caregivers
GROUP BY Availability;
  
```

The Data output tab displays the results of the query:

Availability	NumberOfCaregivers
true	507
false	493

Total rows: 2 of 2 Query complete 00:00:00.045 Successfully run. Total query runtime: 45 msec. 2 rows affected.

```

SELECT
    Availability,
    COUNT(*) AS NumberOfCaregivers
FROM
    Caregivers
GROUP BY Availability;
  
```

56. The cost range for each type of service, aiding in pricing strategy and market positioning.

ANS:-

```

SELECT
  ServiceType,
  AVG(Cost) AS AverageCost,
  MIN(Cost) AS MinCost,
  MAX(Cost) AS MaxCost
FROM
  Services
GROUP BY ServiceType;

```

ServiceType	averagecost	mincost	maxcost
Appliance Repair	72.307031707	23.14	119.58
Cleaning	72.274183678	20.33	119.94
Carpentry	67.3127380952	21.15	119.59
Plumbing	71.3597577893	20.08	118.92
Haircuts	71.2487222222	20.10	119.64
Massage Therapy	82.6924170616	22.33	131.89
Photographers	80.7100000000	80.71	80.71
Yoga	77.8800000000	52.37	103.44
Beauty Treatments	69.9185185185	20.14	119.87
Painting	69.9380000000	22.81	109.99

Total rows: 10 of 10 Query complete 00:00:00.071 Successfully run. Total query runtime: 71 msec. 10 rows affected.

```

SELECT
  ServiceType,
  AVG(Cost) AS AverageCost,
  MIN(Cost) AS MinCost,
  MAX(Cost) AS MaxCost
FROM
  Services
GROUP BY ServiceType;

```

57. The distribution of experience among caregivers, which is essential for service quality assessment.

ANS :-

```

SELECT
    Experience,
    COUNT(*) AS NumberOfCaregivers
FROM
    Caregivers
GROUP BY Experience
ORDER BY Experience;

```

The screenshot shows the pgAdmin 4 interface. The left pane is the Browser window displaying database objects like Schemas, Tables, and Functions. The main pane is the Query Editor with the above SQL query. Below the query is the Data output tab showing the results:

Experience	NumberOfCaregivers
1	203
2	181
3	206
4	210
5	200

Total rows: 5 of 5 Query complete 00:00:00.045 Successfully run. Total query runtime: 45 msec. 5 rows affected.

SELECT

Experience,

COUNT(*) AS NumberOfCaregivers

FROM

Caregivers

GROUP BY Experience

ORDER BY Experience;

58. insights into client demographics, language preferences, and their engagement level in terms of message interactions.

ANS :-

```

pgAdmin 4
File Object Tools Help
Browser Dashboard Properties SQL Statistics Dependencies Dependents postgres/post... postgres/postgres@PostgreSQL 14*
Query Open History
1 SELECT
2   Name,
3   DOB,
4   PreferredLanguage,
5   COUNT(M.messageID) AS TotalMessages
6 FROM
7   Client C
8 LEFT JOIN Messages M ON C.ClientID = M.caregiverid
9 GROUP BY C.ClientID;
10

```

	name	dob	preferredlanguage	totalmessages
1	Jenny Wagon	2018-01-23	German	2
2	Phyllis Fleckinbaum	2011-07-27	German	1
3	Gerome Pozzo	2002-02-09	Spanish	4
4	Fulvia Rival	2012-07-10	French	2
5	Shirine Bottrell	2009-09-26	Spanish	0
6	Desiree Scarlett	2012-07-07	Spanish	1
7	Haskell Tippett	2022-06-26	Spanish	0
8	Georg Kensi	2003-01-17	Spanish	1
9	Basilus Roedl	2008-05-05	Spanish	2
10	Parry Bellow	2002-09-16	English	1
11	Bartolomeo Lifford	2011-01-22	French	0
12	Noselyn Bills	2014-03-23	French	2
..
Total rows: 1000 of 1000	Query complete 00:00:00.056			

```

SELECT
  Name,
  DOB,
  PreferredLanguage,
  COUNT(M.messageID) AS TotalMessages
FROM
  Client C
LEFT JOIN Messages M ON C.ClientID = M.caregiverid
GROUP BY C.ClientID;

```

59 . Details about the contact information of clients, which is vital for communication and service delivery.

ANS :-

```

SELECT
  C.Name,
  CC.email,
  CC.phone,
  CC.address
FROM
  ClientContacts CC
JOIN Client C ON CC.ClientID = C.ClientID;

```

	name	email	phone	address
1	Philomena Rivero	lrogen1@gov.net	+20 218 631 8680	3 Grim Pass
2	Audi Manger	gyse@engr.berkeley.edu	+232 328 980 7396	639 Randy Hill
3	Georg Hartigan	ctenneow0@multiplay.c...	+89 122 530 8420	2 Service Plaza
4	Munroe Rowberry	ysarnto1@washington...	+62 558 157 9357	2396 Chinook Road
5	Cassandra Neeseen	rxleoch2@youca...	+7 966 513 4933	6549 Linden Court
6	Anilese Benocchi	cbradbrook3@123-reg...	+81 180 490 8846	46 Oakridge Pass
7	Austin Matlys	jennau4@japanpost.jp	+81 400 179 1911	30 Waxing Park
8	Aura Orkney	stayer@guic.edu	+62 569 672 3758	45937 Tomcat Trail
9	Coralyn Crofts	jmasenegg@shotdot.o...	+63 111 786 3344	9 Quincy Street
10	Benoit Marchent	mwendan7@about.me	+62 653 801 0912	64859 Sheridan Junction
11	Ainslee Farney	tshaney@yolaosite.com	+93 789 921 7730	17 Carey Court
12	Nardi Littimore	nouggypcworld.com	+55 758 634 4831	864 Burning Wood Plaza

Total rows: 1000 of 1000 Query complete 00:00:00.111 Successfully run. Total query runtime: 111 msec. 1000 rows affected.

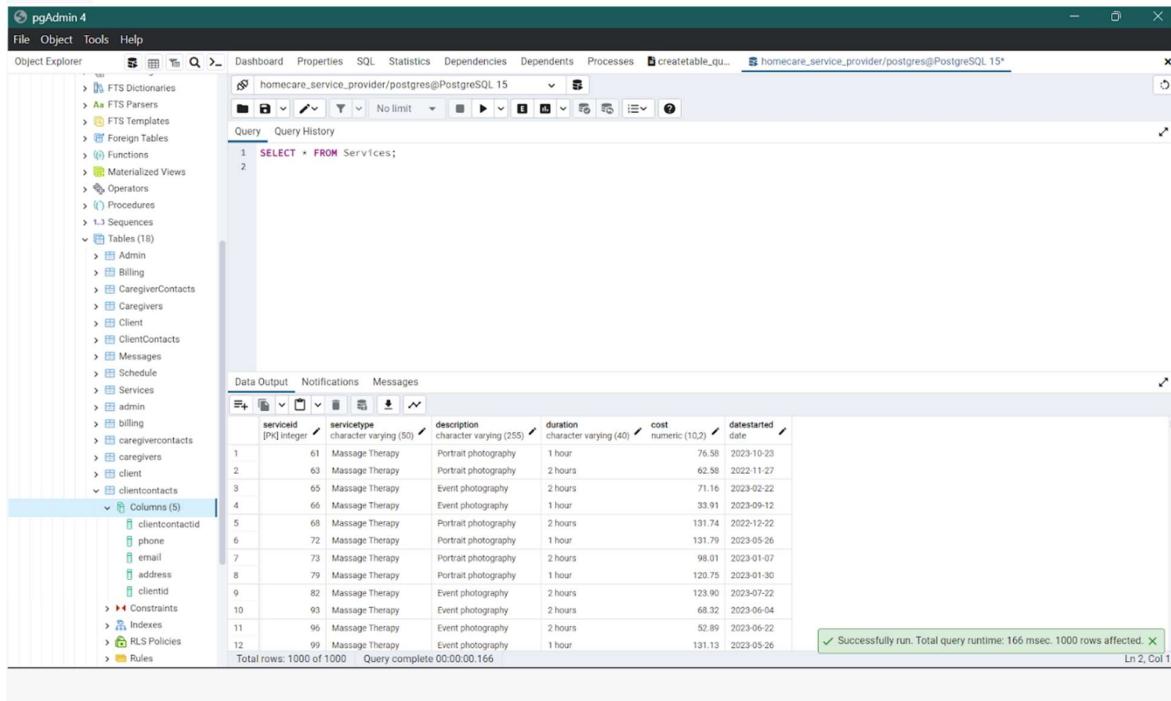
```

SELECT
  C.Name,
  CC.email,
  CC.phone,
  CC.address
FROM
  ClientContacts CC
JOIN Client C ON CC.ClientID = C.ClientID;

```

60.List All Services: Retrieve a list of all services available.

ANS :-



The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with various objects like FTS Dictionaries, Functions, Procedures, Sequences, and Tables (18).
- Query Editor:** Contains the SQL query: `SELECT * FROM Services;`
- Data Output:** Displays the results of the query in a tabular format.
- Table Headers:** `servicid`, `servicetype`, `description`, `duration`, `cost`, `datestarted`.
- Table Data:** 12 rows of service information.
- Message Bar:** Shows "Successfully run. Total query runtime: 166 msec. 1000 rows affected." and "Ln 2, Col 1".

servicid	servicetype	description	duration	cost	datestarted
1	61	Massage Therapy	Portrait photography	1 hour	76.58 2023-10-23
2	63	Massage Therapy	Portrait photography	2 hours	62.58 2022-11-27
3	65	Massage Therapy	Event photography	2 hours	71.16 2023-02-22
4	66	Massage Therapy	Event photography	1 hour	33.91 2023-09-12
5	68	Massage Therapy	Portrait photography	2 hours	131.74 2022-12-22
6	72	Massage Therapy	Portrait photography	1 hour	131.79 2023-05-26
7	73	Massage Therapy	Portrait photography	2 hours	98.01 2023-01-07
8	79	Massage Therapy	Portrait photography	1 hour	120.75 2023-01-30
9	82	Massage Therapy	Event photography	2 hours	123.90 2023-07-22
10	93	Massage Therapy	Event photography	2 hours	68.32 2023-06-04
11	96	Massage Therapy	Event photography	2 hours	52.89 2023-06-22
12	99	Massage Therapy	Event photography	1 hour	131.13 2023-05-26

```
SELECT * FROM Services;
```

61. Find Customers without Allergies: Retrieve customers who have no allergies listed in their medical history.

ANS :-

```

SELECT Name AS CustomerName, MedicalHistory
FROM client
WHERE MedicalHistory NOT LIKE '%allergy%';

```

customername	medicalhistory
Johannah Veldens	Diabetes
Charlot Coen	Hypertension
Perl Tesoe	Diabetes
Timmie Graybeal	Diabetes
Raychel Nutt	Asthma
Adai Robun	Hypertension
Kermie Lame	Asthma
Hussen Dunseath	Diabetes
Darleen Britzman	Diabetes
Betteann Mitchinson	Hypertension
Dallis Alvarado	Diabetes
Carson Rowbrey	Diabetes

Successfully run. Total query runtime: 86 msec. 872 rows affected. Ln 2, Col 12

```

SELECT Name AS CustomerName, MedicalHistory
FROM Customers
WHERE MedicalHistory NOT LIKE '%allergy%';

```

62. Sort Services by Cost in Descending Order: List services sorted by their cost in descending order.

ANS :-

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with various tables like Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, admin, billing, caregivercontacts, caregivers, client, clientcontacts, and columns.
- Query Editor:** Contains the SQL query:


```
1 SELECT * FROM Services
2 ORDER BY cost DESC;
```
- Data Output:** Displays the results of the query, showing 1000 rows of service data. The columns include serviceid, servicetype, description, duration, cost, and datestarted.
- Status Bar:** Shows a message indicating the query was successfully run with a runtime of 127 msec and 1000 rows affected.

```
SELECT * FROM Services
ORDER BY cost DESC;
```

63. Calculate the Total Number of Services: Count the total number of services available.

ANS :-

The screenshot shows the pgAdmin 4 interface with the following details:

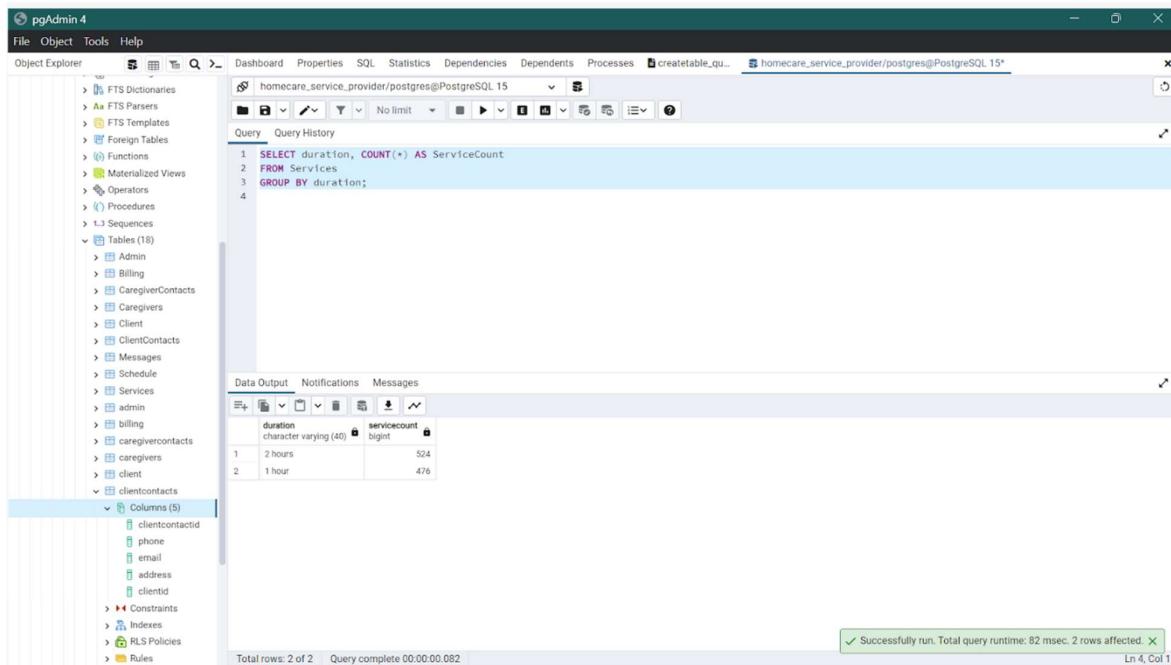
- Object Explorer:** Shows the database schema with various tables like Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, admin, billing, caregivercontacts, caregivers, client, clientcontacts, and columns.
- Query Editor:** Contains the SQL query:


```
1 SELECT COUNT(*) AS TotalServices
2 FROM Services;
3
```
- Data Output:** Displays the results of the query, showing 1 row with the value 1000 under the column 'totalservices'.
- Status Bar:** Shows a message indicating the query was successfully run with a runtime of 66 msec and 1 rows affected.

```
SELECT COUNT(*) AS TotalServices
FROM Services;
```

64. Group Services by Duration: Group services by their duration and calculate the count in each group.

ANS :-



```
pgAdmin 4
File Object Tools Help
Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes
homecare_service_provider/postgres@PostgreSQL 15*
Query Query History
1 SELECT duration, COUNT(*) AS ServiceCount
2 FROM Services
3 GROUP BY duration;
4

Data Output Notifications Messages
duration servicecount
character varying(40) bigint
1 2 hours 524
2 1 hour 476

Total rows: 2 of 2 Query complete 00:00:00.082
Successfully run. Total query runtime: 82 msec. 2 rows affected. Ln 4, Col 1
```

```
SELECT duration, COUNT(*) AS ServiceCount
FROM Services
GROUP BY duration;
```

65. Provides contact details of caregivers, crucial for internal coordination and emergency contact.

ANS:-

```

SELECT
    CG.Name,
    CGC.phone,
    CGC.email,
    CGC.address
FROM
    CaregiverContacts CGC
JOIN Caregivers CG ON CGC.CaregiverID = CG.CaregiverID;

```

	name	phone	email	address
1	Alexina Pogson	+967 536 452 845	lmacombe@gmail.ru	1940 La Follette Parkway
2	Allene Popworth	+86 656 545 3564	yfannin1@siashmidt.org	9 Telisman Point
3	Zoranya Sanchez	+86 480 278 7916	rsoal@gnbua.com	559 Dexter Road
4	Gerty Liddel	+52 975 321 6728	malian13@plannd.org	366 Cordoba Trail
5	Ruthe Thirk	+62 661 200 6078	rlemeij5@spoke.de	48597 Pawling Street
6	Austine Brazenor	+234 486 625 9081	rlemeij5@spoke.com	6126 La Follette Junction
7	Auge Adornos	+50 287 448 6172	sachromonoff@parallels.com	1238 Florence Alley
8	Tarrance Domineaw	+86 372 115 1373	kpuwe7@uniblog.fr	8731 Ridgeview Lane
9	Ninette Estby	+62 708 245 8674	osyclebeck4@gazmodo.com	58763 Fuller Alley
10	Stace Amonorth	+86 223 790 5942	tshetoni9@wp.com	72084 Crowley Terrace
11	Zoranya Sanchez	+976 266 600 6517	mastela@eznearables.com	6730 Shelley Crossing
12	Reidar Groul	+62 664 873 2951	mschwan7@sourceforge.net	91 Mayfield Way

Total rows: 1000 of 1000 Query complete 00:00:00.082 Successfully run. Total query runtime: 82 msec. 1000 rows affected.

```

SELECT
    CG.Name,
    CGC.phone,
    CGC.email,
    CGC.address
FROM
    CaregiverContacts CGC
JOIN Caregivers CG ON CGC.CaregiverID = CG.CaregiverID;

```

66. A detailed view of billing for services, including client and caregiver details, service types, and payment information, offering a comprehensive financial perspective.

ANS :-

```

SELECT
    B.InvoiceID,
    Cl.Name AS ClientName,
    Cg.Name AS CaregiverName,
    Sv.ServiceType,
    B.Amount,
    B.PaymentStatus,
    B.PaymentMethod
FROM
    Billing B
JOIN Schedule S ON B.SID = S.SID
JOIN Client Cl ON S.ClientID = Cl.ClientID
JOIN Caregivers Cg ON S.CaregiverID = Cg.CaregiverID
JOIN Services Sv ON S.ServiceID = Sv.ServiceID;

```

The screenshot shows the pgAdmin 4 interface with a query editor containing the provided SQL code. The results are displayed in a table with the following data:

InvoiceID	clientname	caregivername	servicetype	amount	paymentstatus	paymentmethod
1	Cecily Bea	Xyla Peeling	Haircuts	195.81	Paid	Credit Card
2	Dentrie Inn	Brewer Hemphill	Haircuts	65.13	Paid	PayPal
3	Haze Venique	Kacey Hemphill	Beauty Treatments	534.29	Unpaid	PayPal
4	Benjamen Philpot	Page Dubois	Beauty Treatments	208.00	Paid	PayPal
5	Wood Pacey	Bobine Balach	Massage Therapy	216.79	Unpaid	Cash
6	Ceciley Zierne	Davine Ladd	Cleaning	164.03	Paid	Credit Card
7	Parisif Ivison	Tylie Dasi	Massage Therapy	366.79	Unpaid	Cash
8	Kurt Haney'	Pivot Linham	Cleaning	363.44	Unpaid	Credit Card
9	Dela Terri	Sarajane Fender	Massage Therapy	274.48	Paid	Cash
10	Dominic Bendorick	Marcella Foulke	Beauty Treatments	147.87	Unpaid	PayPal
11	Lulu Greenhill	Geri Wardell	Haircuts	365.70	Paid	PayPal
12	Edward Roskam	Beakie Brooks	Appliance Repair	174.52	Unpaid	Cash

Total rows: 1000 of 1000 Query complete 00:00:00.059

Successfully run. Total query runtime: 59 msec. 1000 rows affected.

```

SELECT
    B.InvoiceID,
    Cl.Name AS ClientName,
    Cg.Name AS CaregiverName,
    Sv.ServiceType,
    B.Amount,
    B.PaymentStatus,
    B.PaymentMethod
FROM
    Billing B
JOIN Schedule S ON B.SID = S.SID
JOIN Client Cl ON S.ClientID = Cl.ClientID
JOIN Caregivers Cg ON S.CaregiverID = Cg.CaregiverID
JOIN Services Sv ON S.ServiceID = Sv.ServiceID;

```

67. Count Unique Service Types: Count the number of unique service types available.

ANS:-

The screenshot shows the pgAdmin 4 interface for PostgreSQL 15. The left sidebar displays the Object Explorer with various database objects like FTS Dictionaries, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, Tables (18), and their respective columns and constraints. The main pane shows a query editor with the following SQL code:

```
1 SELECT COUNT(DISTINCT servicetype) AS UniqueServiceTypes
2 FROM Services;
3
```

The Data Output tab shows the result of the query:

unique servicetypes	bignum
1	10

At the bottom, status messages indicate "Successfully run. Total query runtime: 77 msec. 1 rows affected." and "Ln 3, Col 1".

```
SELECT COUNT(DISTINCT servicetype) AS UniqueServiceTypes  
FROM Services;
```

68. The roles of admin staff in managing caregivers and services, indicating their workload and areas of responsibility.

ANS :-

The screenshot shows the pgAdmin 4 application window. The left sidebar contains a tree view of database objects under the 'Browser' tab, including Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas, hawd, public, Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables. The 'Tables' node is currently selected.

The main area has tabs for Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, and a connection status bar for 'postgres/postgres@PostgreSQL 14*'. A query editor window is open with the following SQL query:

```
1 SELECT
2     A.Name AS AdminName,
3     A.Role,
4     COUNT(DISTINCT C.CaregiverID) AS CaregiversOverseen,
5     COUNT(DISTINCT S.ServiceID) AS ServicesManaged
6 FROM
7     Admin A
8     JOIN Schedule S ON A.AdminID = S.CaregiverID
9     JOIN Caregivers C ON S.CaregiverID = C.CaregiverID
10    GROUP BY A.AdminID;
11
```

The 'Data output' tab is active, displaying a table with the results of the query:

	adminname	role	caregiversoverseen	servicesmanaged
1	Berlin Pennuzzi	Senior Developer	1	1
2	Thomie Bagelot	Software Test Engineer	1	2
3	Joyce Gazzetti	Legal Assistant	1	3
4	Lonna Speller	Dental Hygienist	1	1
5	Rey Dochoon	Chief Design Engineer	1	1
6	Ida Branton	Senior Editor	1	1
7	Everett Morefield	Account Executive	1	1
8	Gaspard Wissotsky	Nuclear Power Engineer	1	2
9	Thaddeus Cauley	Technical Writer	1	2
10	Jany Danetti	Senior Financial Analyst	1	2
11	Joanne Whittemore	Editor	1	1
12	Vikky Yegorov	Operator	1	2

Total rows: 627 of 627 Query complete 00:00:00.077

A message bar at the bottom right indicates: Successfully run. Total query runtime: 77 msec. 627 rows affected.

```

SELECT
  A.Name AS AdminName,
  A.Role,
  COUNT(DISTINCT C.CaregiverID) AS CaregiversOverseen,
  COUNT(DISTINCT S.ServiceID) AS ServicesManaged
FROM
  Admin A
JOIN Schedule S ON A.AdminID = S.CaregiverID
JOIN Caregivers C ON S.CaregiverID = C.CaregiverID
GROUP BY A.AdminID;

```

69. List Services with a Cost Less Than 50: Retrieve services with a cost less than 50.

ANS :-

serviceid	servicetype	description	duration	cost	datestarted
1	66	Massage Therapy	Event photography	1 hour	33.91 2023-09-12
2	152	Massage Therapy	Portrait photography	1 hour	24.42 2023-04-06
3	161	Massage Therapy	Portrait photography	2 hours	29.08 2023-08-07
4	183	Massage Therapy	Portrait photography	1 hour	41.69 2023-06-15
5	184	Massage Therapy	Event photography	1 hour	33.55 2023-04-12
6	216	Massage Therapy	Portrait photography	1 hour	35.93 2022-11-30
7	217	Massage Therapy	Portrait photography	2 hours	47.29 2022-11-20
8	232	Massage Therapy	Event photography	1 hour	26.77 2023-03-14
9	248	Massage Therapy	Portrait photography	2 hours	35.70 2022-12-06
10	255	Massage Therapy	Portrait photography	1 hour	36.17 2023-02-07
11	303	Massage Therapy	Portrait photography	2 hours	36.01 2023-01-20
12	309	Massage Therapy	Portrait photography	1 hour	25.26 2023-04-11

```

SELECT * FROM Services
WHERE cost < 50;

```

70. Find Customers with Addresses Containing 'Main St': Retrieve customers who live on streets with 'Main St' in their address.

ANS :-

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like FTS Dictionaries, Functions, Procedures, Tables (18), and Columns (5). The main area is the Query Editor with the following SQL code:

```

1 SELECT Name AS CustomerName, Address
2 FROM client, clientcontacts
3 WHERE Address LIKE '%Main St%';
4

```

The Data Output tab shows the results of the query:

customername	address
character varying (255)	character varying (255)

Total rows: 0 of 0 Query complete 00:00:00.045 Ln 4, Col 1

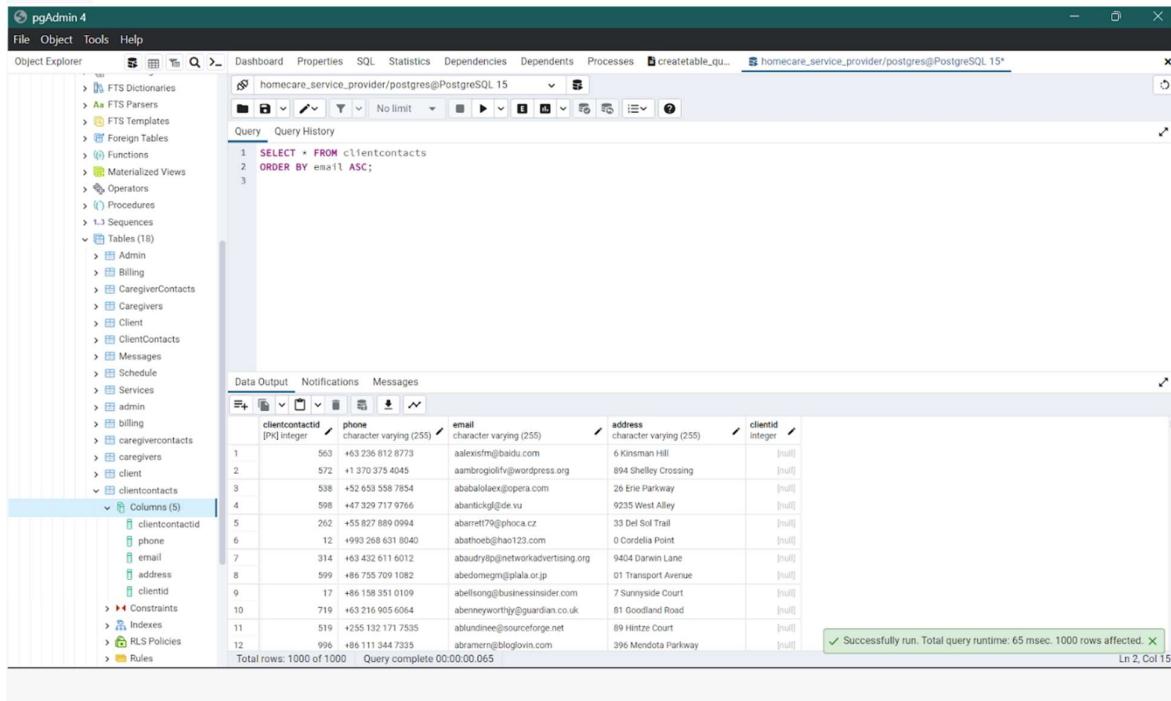
```

SELECT Name AS CustomerName, Address
FROM Customers
WHERE Address LIKE '%Main St%';

```

71. Sort Customers by Contact Number in Ascending Order: List customers sorted by their contact numbers in ascending order.

ANS :-



The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like FTS Dictionaries, Functions, Tables, and Views. The main area is a query editor window titled 'Query' with the following SQL code:

```

1 SELECT * FROM clientcontacts
2 ORDER BY email ASC;
3

```

The results pane shows a table with 12 rows of data from the 'clientcontacts' table. The columns are: clientcontactid, phone, email, address, and clientid. The data includes various email addresses and addresses from around the world. A message at the bottom right of the results pane says "Successfully run. Total query runtime: 65 msec. 1000 rows affected. Ln 2, Col 15".

```

SELECT * FROM clientcontacts
ORDER BY email ASC;

```

72. Count Customers Born Before 1990: Count the number of customers born before the year

1990.

ANS :-

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer tree, which includes FTS Dictionaries, FTS Parsers, FTS Templates, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (18). Under Tables, there are Admin, Billing, CaregiverContacts, Caregivers, Client, ClientContacts, Messages, Schedule, Services, admin, billing, caregivercontacts, caregivers, client, and clientcontacts. The clientcontacts node is expanded, showing Columns (5) with clientcontactid, phone, email, address, and clientid. There are also Constraints, Indexes, RLS Policies, and Rules. The main pane shows a query window with the following SQL code:

```

1 SELECT COUNT(*) AS CustomersBornBefore1990
2 FROM client
3 WHERE DOB < '2013-01-01';
4

```

The Data Output tab shows the results of the query:

	customersbornbefore1990
1	454

Below the results, a message bar indicates: "Successfully run. Total query runtime: 63 msec. 1 rows affected. Ln 4, Col 1". At the bottom of the pgAdmin window, it says "Total rows: 1 of 1 Query complete 00:00:00.063".

```

SELECT COUNT(*) AS CustomersBornBefore1990
FROM Customers
WHERE DOB < '2013-01-01';

```

73. Group Services by Service Type and Count: Group services by service type and count the number of each service type.

ANS :-

The screenshot shows the pgAdmin 4 interface. In the Object Explorer on the left, there are several database objects listed under 'Tables (18)'. In the central 'Query' window, the following SQL code is run:

```

1 SELECT servicetype, COUNT(*) AS ServiceCount
2 FROM Services
3 GROUP BY servicetype;
4 
```

The 'Data Output' tab displays the results of the query:

servicetype	servicecount
Appliance Repair	41
Cleaning	196
Carpentry	84
Plumbing	161
Haircuts	180
Massage Therapy	211
Photographers	1
Yoga	2
Beauty Treatments	108
Painting	16

Total rows: 10 of 10 Query complete 00:00:00.108

Message bar at the bottom right: Successfully run. Total query runtime: 108 msec. 10 rows affected. Ln 4, Col 1

```

SELECT servicetype, COUNT(*) AS ServiceCount
FROM Services
GROUP BY servicetype;
    
```

74. List Services with a Duration of '2 hours': Retrieve services with a duration of '2 hours'.

ANS :-

```

1 SELECT * FROM Services
2 WHERE duration = '2 hours';
3

```

servicid	servicetype	description	duration	cost	datestarted
1	63	Massage Therapy	Portrait photography	2 hours	62.58 2022-11-27
2	65	Massage Therapy	Event photography	2 hours	71.16 2023-02-22
3	68	Massage Therapy	Portrait photography	2 hours	131.74 2022-12-22
4	73	Massage Therapy	Portrait photography	2 hours	98.01 2023-01-07
5	82	Massage Therapy	Event photography	2 hours	123.90 2023-07-22
6	93	Massage Therapy	Event photography	2 hours	68.32 2023-06-04
7	96	Massage Therapy	Event photography	2 hours	52.89 2023-06-22
8	104	Massage Therapy	Event photography	2 hours	111.97 2023-07-09
9	114	Massage Therapy	Portrait photography	2 hours	121.24 2023-04-18
10	116	Massage Therapy	Portrait photography	2 hours	90.67 2023-01-14
11	124	Massage Therapy	Portrait photography	2 hours	130.96 2023-09-13
12	153	Massage Therapy	Portrait photography	2 hours	80.51 2023-10-04

Total rows: 524 of 524 Query complete 00:00:00.113

Successfully run. Total query runtime: 113 msec. 524 rows affected. Ln 3, Col 1

```

SELECT * FROM Services
WHERE duration = '2 hours';

```

75. Calculate the Average Cost of Services: Calculate the average cost of all services

ANS :-

```

1 SELECT AVG(cost) AS AverageCost;
2 FROM Services;
3

```

averagecost
73.45779000000000

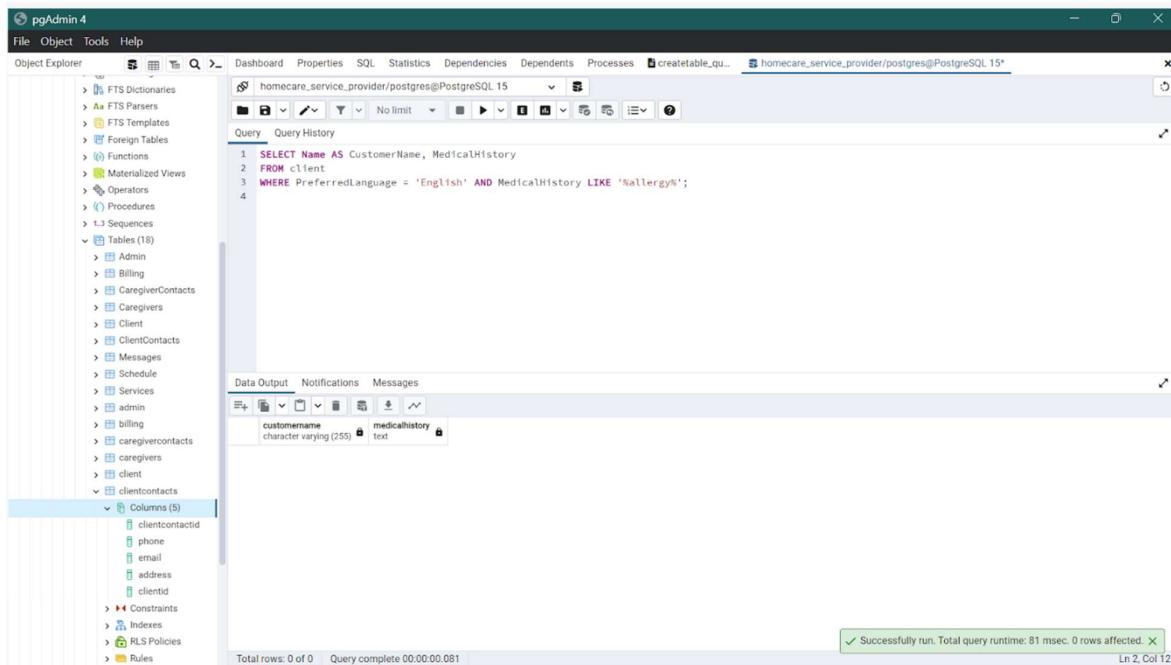
Total rows: 1 of 1 Query complete 00:00:00.091

Successfully run. Total query runtime: 91 msec. 1 rows affected. Ln 3, Col 1

```
SELECT AVG(cost) AS AverageCost
FROM Services;
```

76. Find Customers Who Prefer 'English' and Have Allergies: Retrieve customers who prefer 'English' and have allergies listed in their medical history.

ANS :-



The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like FTS Dictionaries, Functions, Materialized Views, Operators, Procedures, Sequences, and multiple Tables (18). The main pane is a query editor window titled 'homecare_service_provider/postgres@PostgreSQL 15'. It contains the following SQL code:

```
1 SELECT Name AS CustomerName, MedicalHistory
2 FROM client
3 WHERE PreferredLanguage = 'English' AND MedicalHistory LIKE '%allergy%';
4
```

The 'Data Output' tab is selected, showing the results of the query. The output table has two columns: 'customername' (character varying (255)) and 'medicalhistory' (text). There are no rows displayed. At the bottom right of the query window, there is a green success message: 'Successfully run. Total query runtime: 81 msec. 0 rows affected.' and 'Ln 2, Col 12'.

```
SELECT Name AS CustomerName, MedicalHistory
FROM Customers
WHERE PreferredLanguage = 'English' AND MedicalHistory LIKE '%allergy%';
```

77. How each admin is involved in scheduling services and their level of direct engagement with clients.

ANS:-

```

SELECT
  A.Name AS AdminName,
  A.Role,
  COUNT(DISTINCT Sc.SID) AS ScheduledServices,
  COUNT(DISTINCT M.caregiverid) AS EngagedCaregiver
FROM
  Admin A
JOIN Schedule Sc ON A.AdminID = Sc.CaregiverID
LEFT JOIN Messages M ON A.AdminID = M.AdminID
GROUP BY A.AdminID;
  
```

The screenshot shows the pgAdmin 4 interface with a query window displaying the results of the provided SQL query. The results are as follows:

	adminname	role	scheduledservices	engagedcaregiver
1	Bern Penzuu	Senior Developer	1	1
2	Thomie Bagelot	Software Test Engineer..	2	2
3	Joyce Gobato	Legal Assistant	3	3
4	Lonna Spillie	Dental Hygienist	1	2
5	Rey Doonan	Chief Design Engineer	1	0
6	Ida Brauton	Senior Editor	1	0
7	Evelyn Morefield	Account Executive	1	1
8	Gaspard Winstanley	Nuclear Power Engineer	2	0
9	Thadeus Cauley	Technical Writer	2	3
10	Jany Demetis	Senior Financial Analyst	2	1
11	Jo-anne Whittaker	Editor	1	1
12	Vikky Yegorov	Operator	2	2

Total rows: 627 of 627 Query complete 00:00:00.098

```

SELECT
  A.Name AS AdminName,
  A.Role,
  COUNT(DISTINCT Sc.SID) AS ScheduledServices,
  COUNT(DISTINCT M.caregiverid) AS EngagedCaregiver
FROM
  Admin A
JOIN Schedule Sc ON A.AdminID = Sc.CaregiverID
LEFT JOIN Messages M ON A.AdminID = M.AdminID
GROUP BY A.AdminID;
  
```

78. Count Customers with No Emergency Contacts: Count the number of customers who have no emergency contacts specified.

ANS :-

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like FTS Dictionaries, Functions, Tables, and Columns. The main area is the Query Editor with the following SQL code:

```

1 SELECT COUNT(*) AS CustomersWithNoEmergencyContacts
2 FROM client
3 WHERE EmergencyContacts IS NULL;
4

```

The Data Output tab shows the results of the query:

	customerswithnoemergencycontacts
1	0

Below the table, a status bar indicates: Total rows: 1 of 1 | Query complete 00:00:00.075 | Successfully run. Total query runtime: 75 msec. 1 rows affected. | Ln 3, Col 33

```

SELECT COUNT(*) AS CustomersWithNoEmergencyContacts
FROM Customers
WHERE EmergencyContacts IS NULL;

```

79. Find Customers Born in 2000: Retrieve customers who were born in the year 2000.

ANS :-

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like FTS Dictionaries, Functions, Tables, and Views. The main area is a query editor window titled 'Query' with the following SQL code:

```
1 SELECT Name AS CustomerName, DOB
2 FROM Client
3 WHERE DOB >= '2000-01-01' AND DOB <= '2000-12-31';
```

Below the code, the 'Data Output' tab is selected, showing a table with four rows of data:

	CustomerName	DOB
1	Julissa Plumbridge	2000-11-18
2	Fabien Yukhin	2000-12-25
3	Sorcha Eykel	2000-12-29
4	Penn Seminon	2000-11-20

At the bottom right of the interface, there is a message box indicating the query was successfully run: "Successfully run. Total query runtime: 52 msec. 4 rows affected." and "Ln 4, Col 1".

```
SELECT Name AS CustomerName, DOB
FROM Client
WHERE DOB >= '2000-01-01' AND DOB <= '2000-12-31';
```