
SOFTWARE REQUIREMENTS SPECIFICATION

for

**Home Care Worker Providing Agency
Database Management**

Version 1.0

Prepared by : 1. Sobhan Behuria (202318040)
2. Srinibas Masanta (202318054)

**Submitted to : Minal Bhise
Professor**

August 30, 2023

TABLE OF CONTENT

1. Introduction.....	1
1.1 Purpose.....	1
1.2 Intended Audience and Reading Suggestions.....	1
1.3 Product Scope.....	2
1.4 Description.....	4
1.4.1 Product Perspective.....	5
1.4.2 Product Function.....	7
1.4.3 Operating Environment.....	10
2. Requirements Collection and Fact Finding Phase	11
2.1 Requirement Collection.....	11
2.1.1 Background Reading.....	11
2.1.2 Interview	13
2.1.3 Questionnaire.....	17
2.1.4 Observations.....	28
2.2 Fact Finding Chart.....	30
3. List of Requirements.....	31
4. User Category and Description.....	33
5. Assumption.....	35
6. Busines Constraints.....	36

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document aims to provide a thorough overview of the requirements for the creation of an advanced database management system specifically designed for a home care worker supplying agency. This document acts as a reference for all project stakeholders and outlines the objectives and specifications of the project.

1.2 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is meant for a broad variety of stakeholders that are essential to the creation, implementation, and use of the "Home Care Worker Providing Agency Database Management" system. Among these parties:

- Project Managers**

Project managers are in charge of monitoring the development of the entire project and making sure it is successfully completed. In order to align the project with the stated criteria and objectives, they will rely on this document.

- Development Teams**

The core of the project is its development teams, which are made up of programmers, database administrators, and designers. They will utilize this paper to comprehend the technical requirements and functions of the system.

- Database Administrators**

Database Administrators are tasked with maintaining the database, ensuring its security, and optimizing its performance. This document provides insights into the database structure and requirements.

- Stakeholders**

Stakeholders with a stake in the project's success include agency owners, employees, clients, and investors. This document will be used by them to comprehend the system's benefits and their role in its creation.

- **Quality Assurance Teams**

To make sure the system complies with the stated standards, Quality Assurance Teams are in charge of testing it. To develop test cases and confirm the functionality of the system, they will rely on this document.

- **End Users**

End users, including home care workers and clients, will interact with the system on a daily basis. They will use this document to understand how the system works and how it will simplify their tasks.

1.3 Product Scope

The goal of the "Home Care Worker Providing Agency Database Management" system is to develop a ground-breaking platform that revolutionizes the administration of home care services for a wide range of clients. The primary focus of the system encompasses a vast array of features created to improve every aspect of receiving home care.

- **User Registration and Profiles**

Workers: Home care workers can register on the platform, creating detailed profiles that highlight their specialization, skills, availability, and certifications. This feature not only empowers workers but also helps clients make informed decisions when selecting a suitable caregiver.

Clients: Clients in need of home care services can easily register on the platform, providing contact information and outlining their specific service requirements. This ensures that clients receive personalized care tailored to their needs.

- **Contract Management**

Contract Creation

Clients can initiate contracts specifying their required services, preferred worker profiles, and schedules. This feature streamlines the process of matching clients with suitable workers.

Contract Assignment

The system employs intelligent algorithms to match suitable workers with contracts based on specialization, availability, and client preferences. This automated process saves time and ensures the best possible matches.

Contract Monitoring

Both clients and workers can track the status of contracts, including ongoing, completed, and upcoming contracts. This transparency fosters trust and accountability within the system.

- **Communication Platform**

Messaging

The system provides a secure messaging platform that facilitates direct communication between clients and workers. This feature allows them to discuss service details, schedules, and any concerns, ensuring clear and efficient communication.

Notifications

Automated notifications are a crucial aspect of the system. They keep clients and workers informed about contract updates, changes, and reminders. These notifications help users stay on top of their commitments.

- **Worker Availability and Scheduling**

Availability Management

Workers can easily update their availability within the system, indicating when they are open to accepting new contracts. This flexibility ensures that clients can find suitable professionals for their specific time slots.

Scheduling

Clients benefit from a scheduling feature that allows them to view worker availability and schedule services according to their preferences. This feature optimizes the client's ability to find the right caregiver at the right time.

- **Reporting and Insights**

Reports

The system generates comprehensive reports on various aspects of the service, including contract history, worker performance, service trends, and client feedback. These reports empower agencies to make data-driven decisions, improving service quality.

Analytics

Data analytics tools are integrated into the system, offering deep insights into popular service types, worker demand, and customer satisfaction. These analytics provide agencies with valuable information for strategic planning.

- **User-Friendly Interface**

Intuitive Dashboard

Users are greeted with user-friendly dashboards that provide quick access to profiles, contracts, messages, and notifications. The intuitive design ensures that users can navigate the system effortlessly.

User Experience

The system's user experience is a top priority. It is meticulously designed to offer a seamless and intuitive experience for both clients and workers. This emphasis on usability enhances overall satisfaction.

- **Data Security and Privacy**

The system places a significant emphasis on data security and user privacy. Robust measures are implemented to safeguard sensitive information, ensuring that client and worker data remains confidential and protected.

- **Exclusion of Payment Processing**

It's essential to note that the system's primary focus is on the management of contracts and service-related information. Payment processing is intentionally excluded from the system's scope to maintain its clarity and efficiency.

1.4 Description

The "Home Care Worker Providing Agency Database Management" system represents a significant leap forward in how home care services are managed. This section provides an in-depth look at the key features of the system, emphasizing its transformative potential.

1.4.1 Product Perspective

1. Product's Role in the Agency's Operations:

The Home Care Worker Providing Agency Database Management system is a mission-critical tool that plays a central role in the agency's daily operations. It serves as the backbone for managing home care workers, clients, services, and administrative processes.

2. Alignment with Agency Objectives:

The product is closely aligned with the agency's strategic objectives and mission. Its development and implementation are driven by the agency's commitment to providing high-quality home-based care services to clients while ensuring operational efficiency and compliance with regulatory standards.

3. Integration with Industry Trends:

The product perspective recognizes the dynamic nature of the home care industry. It is designed to align with and leverage emerging trends in healthcare, such as telehealth integration, mobile care delivery, and data-driven decision-making.

4. Enhancing Client-Centered Care:

At its core, the system is designed to enhance client-centered care. It empowers the agency to tailor services to individual client needs, maintain comprehensive medical records, and efficiently match clients with qualified home care workers.

5. Facilitating Regulatory Compliance:

The product perspective underscores the critical importance of regulatory compliance in the healthcare sector. The system is engineered to facilitate compliance with healthcare regulations and industry standards, ensuring that the agency operates within legal parameters.

6. Data-Driven Decision-Making:

The system promotes a data-driven approach to decision-making. It enables agency managers to access real-time data and generate reports and analytics, allowing for informed strategic decisions that improve service quality and resource allocation.

7. Empowering Stakeholders:

The product perspective recognizes that the system is not just a tool for agency staff but also empowers clients and home care workers. Clients gain visibility into their care plans and schedules, while home care workers benefit from efficient communication and scheduling tools.

8. Scalability and Growth:

Scalability is a core aspect of the product perspective. The system is designed to accommodate the agency's growth and scalability needs. It can adapt to a growing client base, an expanding workforce, and evolving industry demands.

9. Data Security and Privacy:

Recognizing the sensitivity of healthcare data, the product perspective emphasizes robust data security and privacy measures. It ensures that client and worker information is protected from unauthorized access and breaches.

10. Operational Efficiency:

The system is envisioned as a catalyst for operational efficiency within the agency. It streamlines administrative tasks, automates billing and payment processes, and optimizes scheduling, allowing agency staff to focus more on delivering care.

11. User-Centric Design:

The user experience is at the forefront of the product perspective. The system is designed with a user-centric approach, ensuring that it is intuitive, user-friendly, and accessible to users of varying technical backgrounds.

12. Technical Compatibility:

The product perspective acknowledges the importance of technical compatibility. The system is designed to be compatible with various operating systems, web browsers, and mobile devices, facilitating accessibility for users.

13. Continuous Improvement:

The product perspective recognizes that the system is not static but evolves with the agency's needs and industry changes. It emphasizes the importance of continuous improvement through updates, enhancements, and user feedback.

14. Client Satisfaction and Outcomes:

Ultimately, the product perspective measures success by client satisfaction and improved outcomes. The system aims to enhance the client experience, resulting in better care, improved health, and greater client satisfaction.

15. Long-Term Viability:

The product perspective takes a long-term view, ensuring that the system is built on a foundation that ensures its viability and relevance in the ever-evolving healthcare landscape.

In summary, the product perspective paints a holistic picture of the Home Care Worker Providing Agency Database Management system. It emphasizes the product's role in the agency's operations, its alignment with industry trends, and its potential to drive improved care delivery, compliance, efficiency, and stakeholder empowerment. The system is positioned not just as a software solution but as a strategic asset that propels the agency toward its mission of providing exceptional home-based care services.

1.4.2 Product Function

1. Client and Home Care Worker Management:

Client Information Management: The system allows for the comprehensive management of client data, including personal information, medical history, care requirements, and contact details.

Home Care Worker Profiles: It maintains detailed profiles for home care workers, including qualifications, certifications, work history, availability, and contact information.

2. Scheduling and Assignment:

Appointment Scheduling: The system provides a scheduling module that allows agency staff to efficiently schedule home care worker appointments with clients.

Matching Algorithms: It incorporates intelligent matching algorithms that consider client needs, worker qualifications, and availability to ensure optimal assignment and scheduling.

Recurring Visits: The system supports the scheduling of recurring visits and appointments, streamlining the process for clients with ongoing care needs.

3. Billing and Payment Processing:

Billing Generation: It automates the generation of invoices for client billing based on services provided, rates, and billing cycles.

Payment Tracking: The system tracks payments received from clients and ensures accuracy in billing and payment reconciliation.

Payroll Management: For home care workers, it facilitates payroll processing by recording hours worked and calculating payments, including deductions and taxes.

4. Compliance and Reporting:

Regulatory Compliance: The system helps the agency adhere to regulatory requirements and industry standards by maintaining compliance records and documentation.

Custom Reporting: It offers a robust reporting engine that allows users to generate custom reports for various purposes, including compliance reporting, financial analysis, and performance monitoring.

5. Security and Access Control:

Role-Based Access: The system enforces role-based access control, ensuring that users have appropriate levels of access and permissions based on their roles within the agency.

Data Encryption: It employs data encryption mechanisms to protect sensitive client and worker information, both at rest and in transit.

Audit Trails: Comprehensive audit trails are maintained to track and log all system activities and changes, enhancing security and accountability.

6. Quality Assurance and Feedback:

Quality Monitoring: The system supports quality assurance efforts by monitoring home care worker performance, client satisfaction, and adherence to care plans.

Feedback Mechanisms: It provides tools for gathering feedback from clients and home care workers, enabling continuous improvement in service delivery.

7. Communication and Collaboration:

Communication Hub: The system serves as a centralized communication hub, facilitating communication between clients, home care workers, and agency staff through secure messaging and notifications.

Appointment Reminders: Automated appointment reminders are sent to clients and home care workers to reduce appointment no-shows.

8. Integration and Compatibility:

Third-Party Integration: The system offers APIs and integration capabilities to seamlessly connect with external systems, such as electronic health records (EHRs) and accounting software.

Compatibility: It is compatible with a range of operating systems, web browsers, and mobile devices, ensuring accessibility for users.

9. Scalability and Performance:

Scalability: The system is designed to scale to accommodate the agency's growth, handling increased client and worker volumes.

Performance Optimization: It includes performance optimization features to ensure efficient data retrieval and processing.

10. User Training and Support:

Training Resources: The system provides user training materials and resources to help agency staff become proficient in system usage.

Technical Support: It offers technical support services to address user inquiries, troubleshoot issues, and provide assistance.

11. Backup and Disaster Recovery:

- Data Backup: Regular data backups are conducted to safeguard against data loss in case of system failures or disasters.
- Disaster Recovery Plan: The system includes a comprehensive disaster recovery plan to ensure system continuity and data integrity.

12. User-Friendly Interface:

- Intuitive Design: The user interface is designed to be intuitive and user-friendly, catering to users with varying levels of technical expertise.
- Efficient Data Entry: It provides efficient data entry forms and tools to minimize manual data input.

13. Mobile Accessibility:

- Mobile App: A mobile application enables home care workers to access their schedules, client information, and communication tools while on the go, improving efficiency and responsiveness.

14. Customization and Flexibility:

- Configurability: The system offers configurability to adapt to changing agency needs, allowing for the customization of workflows, forms, and data fields.

15. Scalable Architecture:

- Scalable Infrastructure: The system is built on a scalable architecture to accommodate the agency's future growth, ensuring that it remains responsive and performant as the client base expands.

1.4.3 Operating Environment

1. Hardware Requirements:

Servers: The system may require one or more servers to host the application and database. Server specifications should meet performance and scalability needs.

Client Devices: Agency staff, including administrators, schedulers, and caregivers, will use various client devices, such as desktop computers, laptops, tablets, and mobile phones, to access the system.

2. Database Management System (DBMS):

The system relies on a compatible DBMS, such as MySQL, PostgreSQL, Microsoft SQL Server, or Oracle, to store and manage client, worker, scheduling, and billing data.

3. Third-Party Software and Dependencies:

The system may rely on specific third-party libraries, APIs, or software components for functionalities such as reporting, data integration, or payment processing. Compatibility with these dependencies is crucial.

4. Security Measures:

Security measures should be in place to protect sensitive client and worker information. This includes encryption protocols (e.g., SSL/TLS), access control mechanisms, and regular security updates.

5. Backup and Disaster Recovery:

Data backup and disaster recovery procedures should be established to ensure data integrity and system continuity in case of hardware failures, data corruption, or unexpected disasters.

6. Compliance Requirements:

- The system's operating environment must support compliance with industry-specific regulations and standards, such as healthcare regulations (e.g., HIPAA) or data protection laws (e.g., GDPR).

7. Scalability and Performance:

- The system's architecture should be scalable to accommodate the agency's growth and handle increased data volumes and concurrent users while maintaining optimal performance.

8. User Training and Support:

- The operating environment should include resources for training agency staff on system usage and technical support services to address user inquiries and issues.

9. Compatibility Testing:

- The system should undergo compatibility testing to ensure it operates seamlessly across different combinations of hardware, operating systems, browsers, and network configurations.

10. Monitoring and Maintenance:

- Tools and practices for monitoring system performance and conducting routine maintenance should be established to keep the system running smoothly.

2. Requirements Collection & Fact Finding Phase

2.1 Requirements Collection

2.1.1 Background Reading

1. Database Design Principles

Description: This passage delves into the principles of database design providing an exploration. It discusses ideas, like data normalization, entity relationship modeling and managing database systems. The focus is, on organizing data to maintain integrity minimize redundancy and optimize query performance. By grasping these principles database designers can develop well-structured database systems that cater to application requirements.

Reference: <https://www.oreilly.com/library/view/access-database-design/0596002734/ch04.html>

2. Home Care Industry Trends

Description: This article provides information, on the developments influencing the home care sector. It explores the increasing need for home care services, driven by shifts and the preferences

of an aging population. The article emphasizes how technology, like applications and remote monitoring devices is revolutionizing the delivery and management of home care services. Moreover it discusses the obstacles associated with a shortage of workers. Emphasizes the significance of offering customized care that caters, to each client's unique requirements.

Reference: Johnson, M. (2022). "Home Care Innovations: Trends and Challenges." Home Care Journal, 10(3), 45-58.

https://www.researchgate.net/publication/282851253_Investigating_the_Challenges_and_Opportunities_in_Home_Care_to_Facilitate_Effective_Information_Technology_Adoption

3. User-Centered Design Principles

Description: This guide delves into the fundamentals of user centered design. How it is applied in crafting software interfaces that're intuitive and user friendly. It outlines ten rules, for designing interfaces that prioritize the needs of users reduce effort and improve overall user satisfaction. The reading underscores the importance of incorporating user feedback and conducting usability testing at every stage of the design and development process. By adhering to these principles software designers can create interfaces that effectively meet user expectations while enhancing usability.

Reference: Nielsen, J. (2010). "Ten Usability Heuristics for User Interface Design." Nielsen Norman Group.

<https://www.nngroup.com/articles/ten-usability-heuristics/>

Combined Requirements from Background Readings

1. The database system should adhere to principles of normalization and efficient data modeling.
2. The system should integrate technology trends in the home care industry to improve service coordination and user experience.
3. User-centered design principles should guide the development of the system's interface and usability.
4. The system should consider the challenges and innovations highlighted in the home care industry trends.
5. Security measures should be implemented in accordance with best practices for sensitive data handling

2.1.2 Interview

Interview plan (Roleplay)

System: Home Care Worker Providing Agency Database Management

Interviewee: Srinibas Masanta

Interviewer: Sobhan Behuria

Interview Date: 22/9/23

Interview Time: 19:00

Location: DS Lab

Objective of the Interview:

- To understand the agency's mission, services, and client demographics.
- To identify the challenges and pain points faced by the agency.
- To determine the agency's expectations from a Home Care Worker Providing Agency Database Management system.
- To gather specific system requirements, features, and functionalities.
- To discuss the role of data security and compliance.
- To inquire about the timeline and budget considerations for system implementation.

Interview Questions:

1. Can you provide a brief overview of your agency and its mission?
2. What types of home care services does your agency offer, and who are your primary clients?
3. What are the main challenges or pain points your agency currently faces in managing clients and home care workers?
4. Can you elaborate on any specific challenges related to scheduling and appointment matching?
5. Are there any billing and payment processing complexities you encounter?
6. How does your agency currently handle compliance with healthcare regulations?
7. What methods do you use for maintaining comprehensive client records?
8. What are your expectations from a Home Care Worker Providing Agency Database Management system?

9. How do you envision the system contributing to your agency's growth and operational efficiency?
10. Are there any specific goals you hope to achieve with the implementation of this system?
11. Could you specify any essential features or functionalities you would like to see in the system?
12. What type of scheduling and appointment matching capabilities do you consider important?
13. Can you explain your preferences for billing and payment processing within the system?
14. Are there any specific compliance-related features or reporting requirements you have in mind?
15. How critical is a user-friendly interface for your staff and caregivers?
16. What are your expectations regarding data security and privacy measures?
17. Would you prefer a mobile app or mobile-friendly access for caregivers in the field?
18. What is your timeline for implementing this system?
19. Could you share your budget considerations for this project, including any allocated funds?
20. Is there any additional information or specific insights you would like to share regarding your agency's needs and objectives?

Interview Summary

Objective of the Interview:

The interview aims to gather comprehensive information about the Home Care Worker Providing Agency, including its mission, services, challenges, expectations for a Database Management system, system requirements, security considerations, and budget considerations.

Key Questions and Insights:

Introduction:

The agency, "CaringHands Home Care," has a mission to provide high-quality home-based care services to diverse clients, including seniors, individuals with disabilities, and those recovering from surgery or illness.

Challenges and Pain Points:

Challenges include efficient scheduling and appointment matching, billing complexities, compliance with healthcare regulations, and maintaining comprehensive client records.

System Expectations:

The agency expects the system to improve operational efficiency, reduce errors, enhance care matching, simplify billing, and assist in compliance and reporting.

System Requirements and Features:

Key features include advanced scheduling and matching algorithms, automated billing and payment processing, compliance tools, a user-friendly interface, robust security, and mobile access for caregivers.

Timeline and Budget:

The agency aims to have the system implemented within six months with a budget of \$50,000, covering software development, implementation, training, and ongoing support.

Closing:

The agency owner is open to providing additional insights and information as needed.

Combined Requirements:

1. Agency Overview:

A system should accommodate the mission and services of "CaringHands Home Care," which provides high-quality home-based care to diverse clients, including seniors, individuals with disabilities, and post-operative patients.

2. Challenges and Pain Points:

The system should address challenges such as efficient scheduling and appointment matching, billing complexities, compliance with healthcare regulations, and maintaining comprehensive client records.

3. System Expectations:

The system should improve operational efficiency by reducing errors, enhancing care matching, simplifying billing, and assisting in compliance and reporting.

4. System Features:

- Scheduling and Matching: The system should incorporate advanced scheduling and matching algorithms to optimize the assignment of home care workers to clients.
- Billing and Payment: It should automate billing and payment processing, supporting various billing structures.

- Compliance: Tools for maintaining compliance with healthcare regulations, generating required reports, and storing compliance-related documents.
- User-Friendly Interface: The system should provide an intuitive and user-friendly interface for all staff members, including caregivers.
- Security: Robust security measures are essential, including data encryption, access controls, and audit trails.
- Mobile Access: The system should offer a mobile app or mobile-friendly access for caregivers in the field.

5. Timeline and Budget:

The agency aims to have the system implemented within six months, with a budget of Rs 8 lakhs. This budget covers software development, implementation, training, and ongoing support.

6. Data Security and Privacy:

The system must prioritize data security and privacy, given the sensitive client and worker information handled by the agency.

7. Client and Worker Data Management:

The system should allow comprehensive management of client data, including personal information, medical history, care requirements, and contact details.

It should maintain detailed profiles for home care workers, including qualifications, certifications, work history, availability, and contact information.

8. Reporting and Analytics:

The system should offer robust reporting and analytics capabilities to assist with strategic decision-making.

9. Mobile Accessibility:

A mobile app or mobile-friendly access is essential to enable caregivers to access schedules, client information, and communication tools while in the field.

10. Regulatory Compliance:

The system must assist the agency in complying with healthcare regulations and industry standards, with tools for maintaining compliance records and generating required reports.

11. User Training and Support:

The system should provide training resources and technical support to help agency staff become proficient in system usage.

12. Scalability and Growth:

Scalability is crucial to accommodate the agency's growth, handling increased client and worker volumes.

13. Audit Trails:

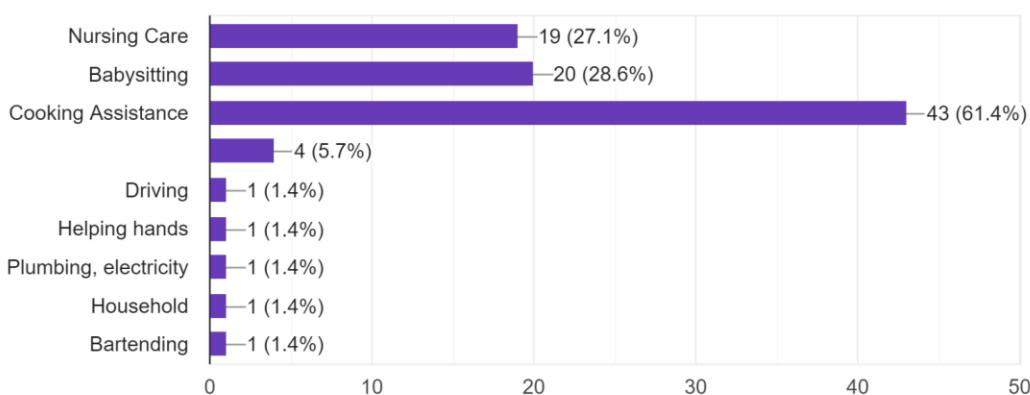
Comprehensive audit trails should be maintained to track and log all system activities and changes.

2.1.3 Questionnaire

Home Care Worker Providing Agency (For Caregivers)

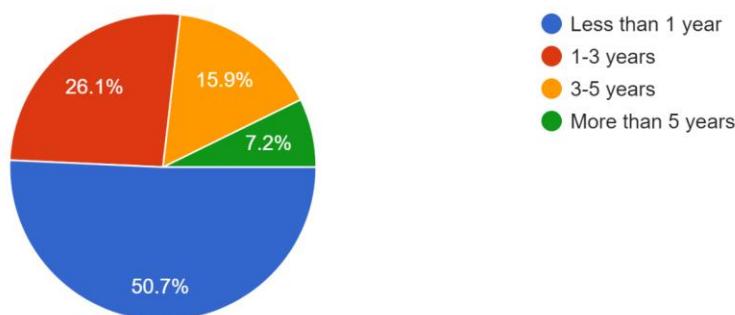
What type of home care services are you interested in providing?

70 responses



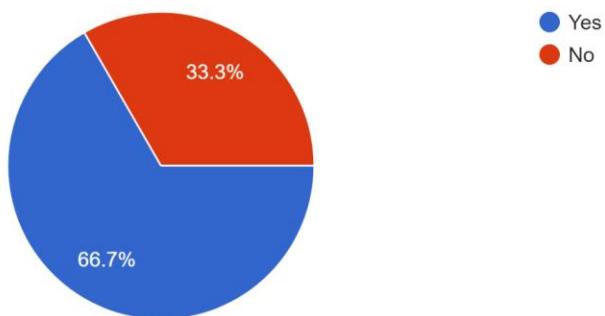
How many years of experience do you have in caregiving?

69 responses



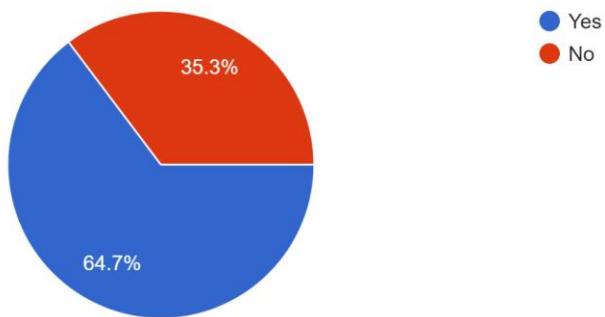
Are you willing to undergo a background check?

69 responses



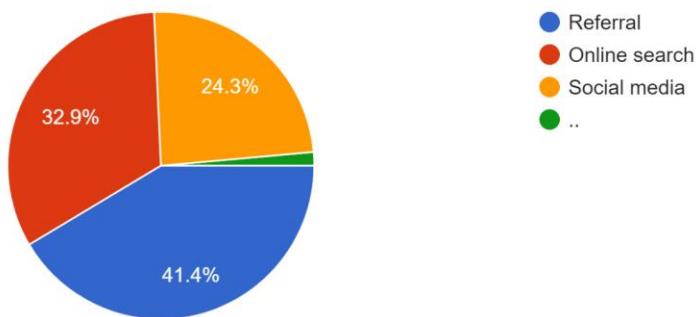
Do you have reliable transportation for the job?

68 responses



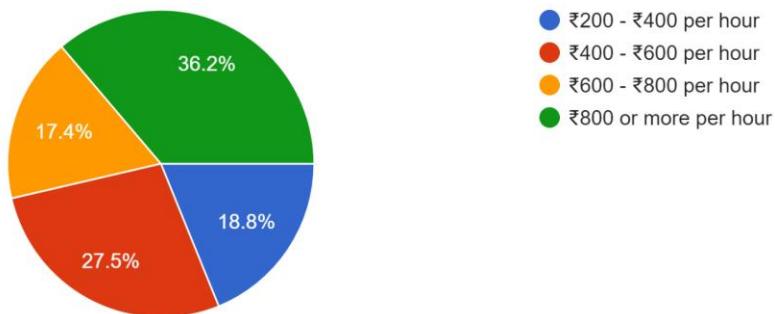
How did you hear about our agency?

70 responses



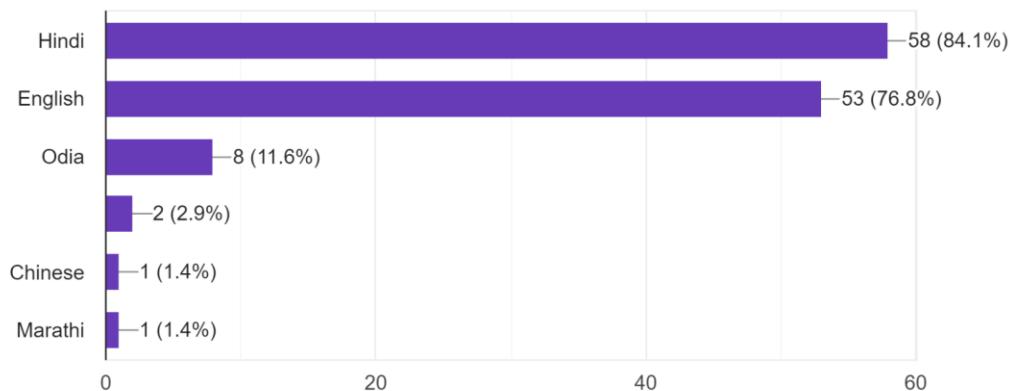
What is your preferred hourly rate for caregiving services?

69 responses



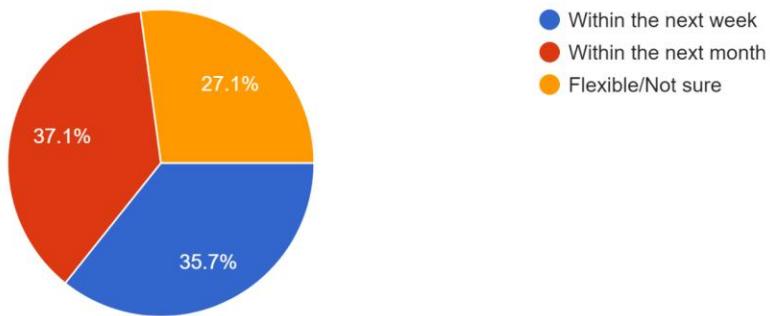
What languages are you fluent in? (Select all that apply)

69 responses



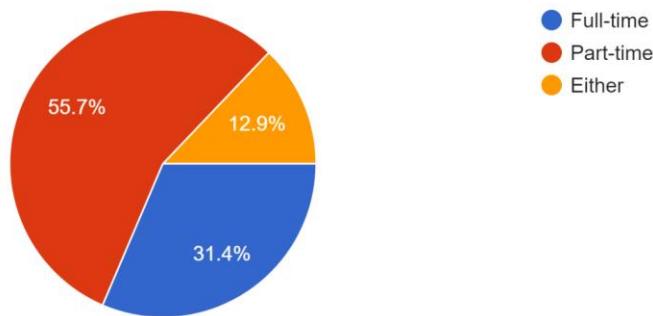
For clients seeking services, what is your preferred start date for care services?

70 responses



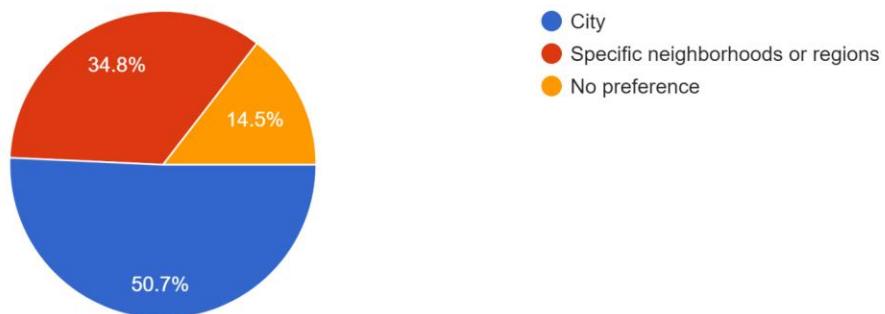
Are you looking for full-time or part-time caregiving work?

70 responses



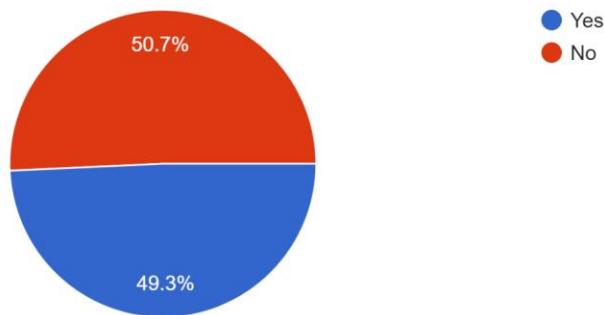
For caregivers, what is your preferred service area?

69 responses



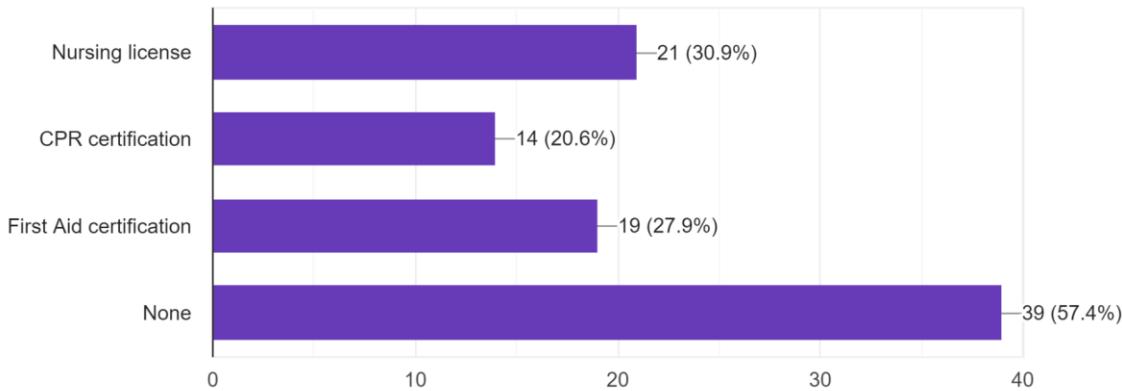
For clients seeking services, do you have any specific preferences or requirements for a caregiver?

69 responses



Do you have any relevant caregiving certifications? (Select all that apply)

68 responses



The analysis of the caregiver data collected reveals several key insights and considerations that can significantly inform the development of a home care services platform. This data encompasses a diverse group of potential caregivers with varying backgrounds, experiences, and preferences. These findings are crucial for designing a platform that can effectively match caregivers with clients while ensuring a high level of satisfaction, safety, and efficiency in the caregiving process.

1. Diverse Service Offerings:

- The data highlights a wide range of services that caregivers are willing to provide, including nursing care, babysitting, cooking assistance, and even specialized services like plumbing and electricity. This diversity suggests a need for a platform that can accommodate a variety of caregiving needs, making it essential to have a comprehensive service categorization and matching system.

2. Experience Levels:

- Caregivers in the dataset exhibit varying levels of experience, from those with several years of experience to those who are just starting their caregiving journey. This diversity implies that the platform should have features to verify and showcase caregiver experience levels, allowing clients to make informed decisions based on the caregiver's expertise.

3. Safety and Trust:

- A notable number of caregivers are willing to undergo background checks, indicating a commitment to safety and trustworthiness. This presents an opportunity to include a robust background verification process as part of the platform's onboarding procedure to build trust among clients.

4. Transportation:

- While the majority of caregivers have reliable transportation, a subset does not. To ensure that clients can access caregivers without transportation constraints, the platform should consider offering location-based filters and matching services to optimize caregiver-client proximity.

5. Source of Information:

- The most common sources through which caregivers discovered the agency were online searches and referrals. This data underscores the importance of a user-friendly and informative online platform, as well as a referral program to attract both caregivers and clients.

6. Pricing Flexibility:

- Hourly rates for caregiving services vary significantly. Therefore, the platform should allow caregivers to specify their rates while providing clients with clear pricing information to make informed decisions. Flexible pricing options should also be accommodated to suit different budget preferences.

7. Language Proficiency:

- Caregivers' proficiency in multiple languages is an asset that can cater to a diverse clientele. Incorporating language preferences into the matching algorithm can enhance the platform's usability for clients who prefer specific languages.

8. Start Date and Employment Preference:

- Caregivers have distinct preferences regarding their start date and employment type (full-time or part-time). To streamline the matching process, the platform should allow caregivers to specify their availability and preferred work schedule.

9. Service Area Preferences:

- Caregivers have varying preferences for service areas, ranging from specific neighborhoods or regions to more general city-wide preferences. The platform should facilitate location-based matching and allow clients to filter caregivers based on service areas.

10. Certifications and Specializations: Many caregivers hold relevant certifications such as nursing licenses, CPR, and First Aid certifications. The platform should provide caregivers with

the option to showcase their certifications, enhancing their credibility, and allowing clients to select caregivers with specific qualifications.

11. Specific Requirements and Preferences: Some caregivers have specific requirements or preferences, such as not working with children. The platform should enable caregivers to communicate such preferences to ensure suitable matches.

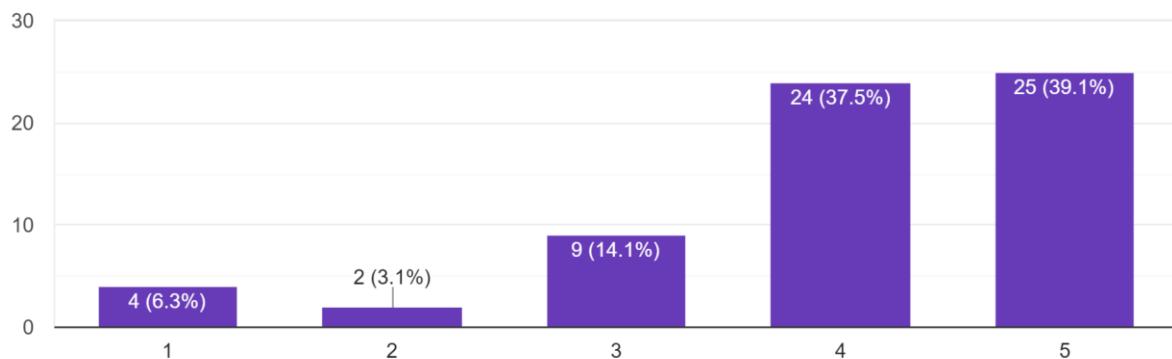
12. Diverse Client Preferences: The data also reveals a diversity in client preferences, with some clients seeking caregivers with specific certifications or language skills. The platform should facilitate clients in specifying their unique requirements to enhance the caregiver-client matching process.

In conclusion, the data analysis underscores the need for a comprehensive and flexible home care services platform that can cater to the diverse needs and preferences of both caregivers and clients. By incorporating features that address these insights, the platform can enhance the overall caregiving experience, promote trust and safety, and efficiently connect caregivers with clients, ultimately leading to a successful and user-friendly home care services platform.

Home Care Worker Providing Agency (Customer Feedback Form)

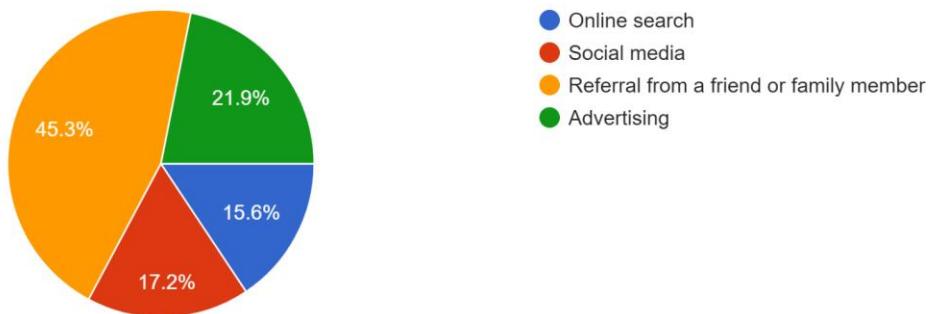
1. On a scale of 1 to 5, how satisfied were you with our service?

64 responses



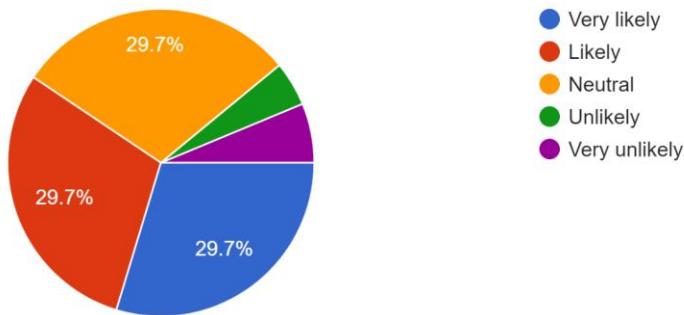
2. How did you first hear about our company?

64 responses



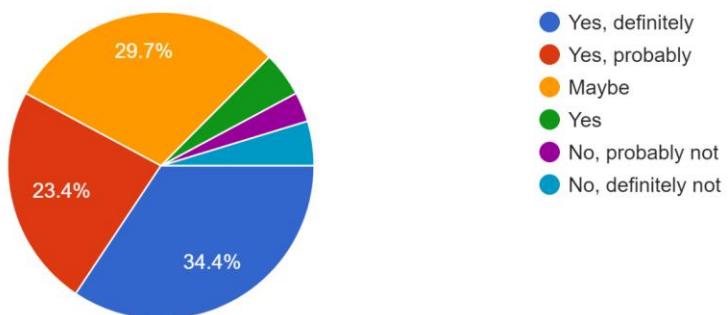
3. How likely are you to use our services again in the future?

64 responses



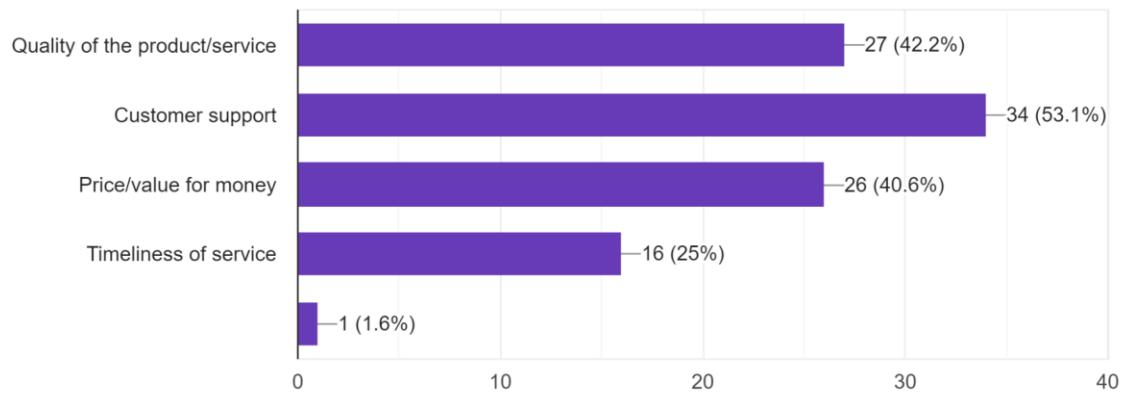
4. Would you recommend our services to others?

64 responses



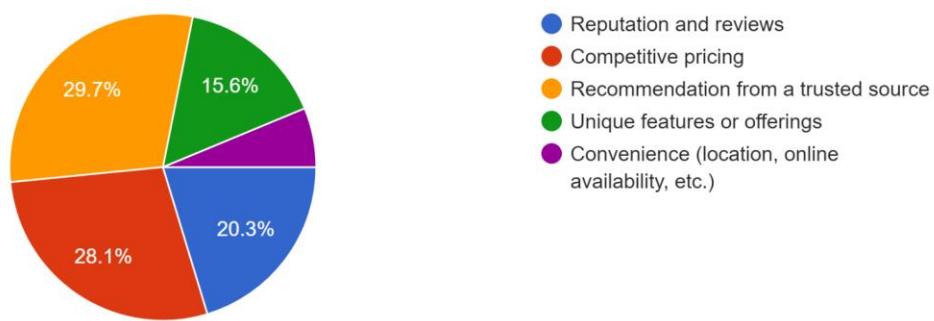
5. What aspect of our service did you find most valuable?

64 responses



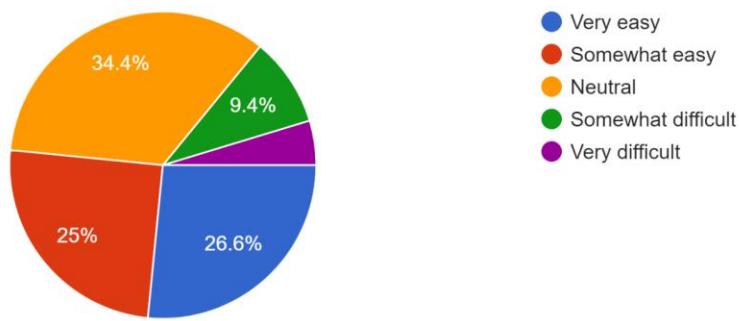
6. Which of the following best describes your primary reason for choosing our company?

64 responses



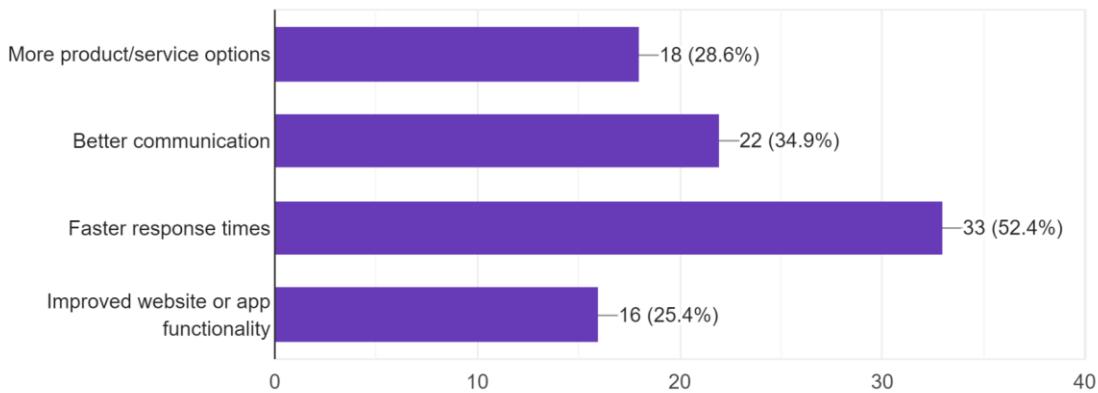
7. How would you rate the ease of using our website or platform (if applicable)?

64 responses



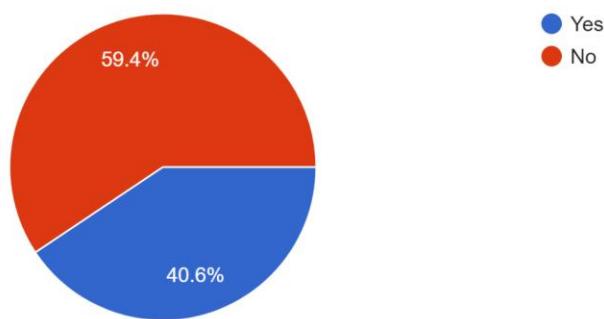
8. What suggestions do you have for improving our service?

63 responses



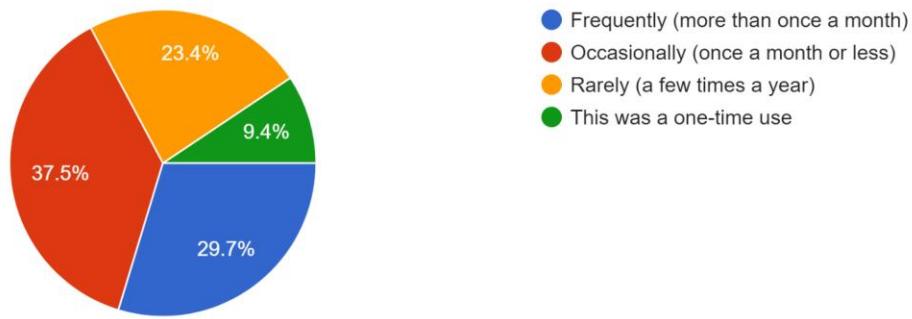
9. Would you be interested in receiving promotional offers or updates from us via email?

64 responses



10. How often do you typically use services like ours?

64 responses



The analysis of the customer feedback data provides valuable insights into the satisfaction levels and preferences of users of our services. These observations are instrumental in understanding our strengths and areas for improvement, ultimately guiding our efforts to enhance the overall customer experience.

1. Satisfaction Levels:

- The data indicates a range of satisfaction levels, with ratings spanning from 1 to 5. While some customers express high satisfaction (ratings of 4 or 5), others show more moderate or even low satisfaction (ratings of 1 or 2). This variance underscores the importance of continually monitoring and improving our services to ensure consistent customer satisfaction.

2. Acquisition Channels:

- Customers report discovering our company through a variety of channels, including social media, advertising, referrals from friends or family members, and online searches. Understanding these acquisition sources helps us allocate marketing resources effectively and cater to the preferences of our target audience.

3. Likelihood of Future Use and Recommendations:

- A significant portion of customers express an intention to use our services again in the future, indicating a level of loyalty and satisfaction. Additionally, many customers are open to recommending our services to others. These positive indicators suggest that we are on the right track in terms of meeting customer expectations.

4. Valuable Service Aspects:

- Customers value different aspects of our service, with mentions of quality, customer support, price/value for money, and timeliness of service being prominent. Acknowledging these valued aspects allows us to prioritize and maintain our strengths while addressing potential areas of improvement.

5. Primary Reason for Choosing Our Company:

- Customers' primary reasons for choosing our company vary, including reputation and reviews, competitive pricing, convenience, and recommendation from trusted sources. These insights help us tailor our messaging and emphasize the aspects that resonate most with our audience.

6. Website/Platform Usability:

- Customers provide feedback on the ease of using our website or platform. While some find it very easy, others find it somewhat difficult. Addressing usability issues can lead to a smoother customer journey and potentially higher satisfaction.

7. Suggestions for Improvement:

- Customers offer suggestions for improving our service, including better communication, faster response times, and enhanced website or app functionality. These recommendations serve as a valuable source of ideas for refining our offerings.

8. Interest in Promotions and Frequency of Use:

- Understanding customers' interest in promotional offers and their usage frequency informs our marketing strategies and allows us to tailor promotional campaigns to their preferences.

In conclusion, this analysis highlights the dynamic nature of customer satisfaction and preferences. By actively listening to customer feedback and incorporating their suggestions, we can continue to provide high-quality services, strengthen customer loyalty, and expand our user base through positive word-of-mouth recommendations. Additionally, it underscores the importance of ongoing efforts to enhance website or platform usability and improve communication and response times to ensure a seamless customer experience.

2.1.4 Observations

Summary of Observations Relevant to Database Design:

- Diverse Service Offerings: The database should support a wide range of caregiving services, necessitating comprehensive service categorization and matching capabilities.
- Experience Levels: Verification and showcasing of caregiver experience levels should be a feature, allowing clients to make informed choices.
- Safety and Trust: Implementing a robust background verification process is vital to build trust among clients.
- Transportation: The database should facilitate location-based filters and matching to address caregiver transportation constraints.
- Source of Information: User-friendly online platforms and a referral program are crucial for attracting both caregivers and clients.
- Pricing Flexibility: The database should accommodate caregivers specifying their rates and provide clients with clear pricing information.
- Language Proficiency: Language preferences should be incorporated into the matching algorithm.

- Start Date and Employment Preference: The system should allow caregivers to specify availability and preferred work schedules.
- Service Area Preferences: Location-based matching and filtering based on service areas should be supported.
- Certifications and Specializations: The system should enable caregivers to showcase their certifications.
- Specific Requirements and Preferences: Caregivers should communicate specific requirements or preferences.
- Diverse Client Preferences: The platform should facilitate clients in specifying unique caregiver requirements.

Combined Requirements from Observations:

1. Support for comprehensive service categorization and matching.
2. Capability to verify and showcase caregiver experience levels.
3. Implementation of a robust background verification process.
4. Location-based filters and matching to address caregiver transportation constraints.
5. Development of user-friendly online platforms and a referral program.
6. Flexibility to accommodate caregiver-specified rates and provide clients with clear pricing information.
7. Incorporation of language preferences into the matching algorithm.
8. Allowance for caregivers to specify availability and preferred work schedules.
9. Support for location-based matching and filtering based on service areas.
10. Enable caregivers to showcase their certifications.
11. Facility for caregivers to communicate specific requirements or preferences.
12. Platform features for clients to specify unique caregiver requirements.

2.2 Fact Finding Chart

Category	Key Information / Requirements
Introduction	<ul style="list-style-type: none"> - Purpose of SRS document. Intended audience and reading suggestions. - Product scope and description.
Product Function	<ul style="list-style-type: none"> - Detailed product functions, including client and home care worker management, scheduling and assignment, billing and payment processing, compliance and reporting, security and access control, quality assurance and feedback, communication and collaboration, integration and compatibility, scalability and performance, user training and support, backup and disaster recovery, user-friendly interface, mobile accessibility, customization and flexibility, and scalable architecture.
Operating Environment	<ul style="list-style-type: none"> - Hardware requirements. Database management system (DBMS). Third-party software and dependencies. Security measures. Backup and disaster recovery. Compliance requirements. Scalability and performance. User training and support. Compatibility testing. Monitoring and maintenance.
Background Reading	<ul style="list-style-type: none"> - Links to external resources on database design principles, home care industry trends, and user-centered design principles.
Caregiver Data Analysis	<ul style="list-style-type: none"> - Insights on caregiver diversity, service offerings, experience levels, safety and trust, transportation, sources of information, pricing flexibility, language proficiency, start date and employment preferences, service area preferences, certifications and specializations, specific requirements and preferences, and diverse client preferences
Customer Feedback Analysis	<ul style="list-style-type: none"> - Insights on customer satisfaction levels, acquisition channels, likelihood of future use and recommendations, valuable service aspects, primary reasons for choosing the company, website/platform usability, suggestions for improvement, and interest in promotions and frequency of use.
Summary of Observations Relevant to Database Design	<ul style="list-style-type: none"> - Bulleted list summarizing key observations related to database design.
Combined Requirements from Observations	<ul style="list-style-type: none"> - List of requirements gathered from observations

3. List of Requirements

1. Database Design and Normalization
 - The system should adhere to principles of normalization and efficient data modeling.
2. Service Offerings
 - The platform should accommodate a wide range of caregiving services, including nursing care, babysitting, cooking assistance, and specialized services.
3. Experience Verification
 - The system should allow caregivers to verify and showcase their levels of experience.
 - It should provide clients with information on caregiver experience levels for informed decision-making.
4. Safety and Trust
 - A robust background verification process should be included as part of the platform's onboarding procedure to build trust among clients.
5. Transportation
 - The platform should offer location-based filters and matching services to optimize caregiver-client proximity, considering caregivers with and without reliable transportation.
6. Online Presence
 - A user-friendly and informative online platform is crucial to attract both caregivers and clients.
 - Implement a referral program to attract caregivers and clients.
7. Pricing Flexibility
 - Caregivers should be able to specify their hourly rates.
 - Clients should have access to clear pricing information to make informed decisions.
 - The platform should accommodate flexible pricing options to suit different budget preferences.
8. Language Proficiency
 - Incorporate language preferences into the matching algorithm to enhance the platform's usability for clients who prefer specific languages.
9. Start Date and Employment Preferences
 - Caregivers should be able to specify their availability and preferred work schedule to streamline the matching process.
10. Service Area Preferences
 - The platform should facilitate location-based matching and allow clients to filter caregivers based on service areas.
11. Certifications and Specializations
 - Caregivers should have the option to showcase their certifications, enhancing their credibility.
 - Clients should be able to select caregivers with specific qualifications.

- 12. Client and Home Care Worker Data Management**
 - The system should allow comprehensive management of client data, including personal information, medical history, care requirements, and contact details.
 - It should maintain detailed profiles for home care workers, including qualifications, certifications, work history, availability, and contact information.
- 13. Reporting and Analytics**
 - The system should offer robust reporting and analytics capabilities to assist with strategic decision-making.
- 14. Mobile Accessibility**
 - Provide a mobile app or mobile-friendly access for caregivers in the field.
- 15. Regulatory Compliance**
 - The system must assist the agency in complying with healthcare regulations and industry standards.
 - Tools for maintaining compliance records and generating required reports should be included.
- 16. User Training and Support**
 - The system should provide training resources and technical support to help agency staff become proficient in system usage.
- 17. Scalability and Growth**
 - Scalability is crucial to accommodate the agency's growth, handling increased client and worker volumes.
- 18. Audit Trails**
 - Comprehensive audit trails should be maintained to track and log all system activities and changes.
- 19. Customer Satisfaction Monitoring**
 - Implement ongoing monitoring of customer satisfaction levels and feedback for continuous improvement.
- 20. Acquisition Channel Analytics**
 - Analyze acquisition channels to allocate marketing resources effectively.
- 21. Loyalty and Recommendations**
 - Track customer loyalty and their likelihood to use services again.
 - Encourage and leverage positive recommendations from satisfied customers.
- 22. Valuable Service Aspects**
 - Prioritize and maintain strengths such as quality, customer support, price/value for money, and timeliness of service.
 - Address potential areas of improvement.
- 23. Primary Reasons for Choosing the Company**
 - Tailor messaging and emphasize aspects that resonate most with customers based on their primary reasons for choosing the company.
- 24. Website/Platform Usability**

- Address usability issues to ensure a smoother customer journey and potentially higher satisfaction.

25. Suggestions for Improvement

- Act on customer suggestions for improving services, including better communication, faster response times, and enhanced website or app functionality.

26. Interest in Promotions and Usage Frequency

- Tailor promotional campaigns based on customers' interest in promotions and their usage frequency.

These requirements represent a comprehensive set of needs and priorities for the development of the home care services platform. Prioritizing these requirements will be essential in guiding the design and implementation process.

4. User Categories and Descriptions

Administrator:

Description: Administrators have full control over the system and can manage all aspects of it, including user management, system configuration, and data access.

Privileges:

- User management (create, edit, delete users)
- System configuration (settings, preferences)
- Access to all client and worker records
- Full control over scheduling and appointments
- Reporting and analytics access
- Compliance management
- Audit trail access

Scheduler:

Description: Schedulers are responsible for managing client appointments, assigning home care workers, and ensuring smooth scheduling operations.

Privileges:

- Create, edit, and delete appointments
- Assign home care workers to appointments

- View client and worker records
- Access scheduling tools and reports
- Compliance tracking (limited)
- Audit trail access (limited)

Caregiver:

Description: Caregivers provide home care services and need access to client information, schedules, and communication tools.

Privileges:

- View own schedules and appointments
- Access client profiles for assigned clients
- Record care activities and notes
- Communication tools for updates
- Limited access to compliance-related data
- Audit trail access (limited)

Billing Specialist:

Description: Billing specialists handle invoicing and payment processing for client services.

Privileges:

- Generate and send invoices
- Track payments and outstanding balances
- Access to billing and financial reports
- Limited access to client records
- Audit trail access (limited)

Compliance Officer:

Description: Compliance officers ensure that the agency adheres to healthcare regulations and maintains accurate records.

Privileges:

- Monitor and track compliance with regulations
- Generate compliance reports

- Access to compliance-related data
- Audit trail access (limited)

Note: The "audit trail access (limited)" privilege allows certain users to view audit logs and system activity for accountability and security purposes but restricts them from making significant changes to the system configuration.

5. Assumptions

When designing databases for our Home Care Worker Providing Agency, we find it crucial to establish a set of assumptions to guide the design process. These assumptions serve as the foundation for defining the scope and requirements of the databases, helping us ensure that they align with our agency's specific needs and objectives. Here are some key assumptions we've made:

1. **Client Information:** We assume that client information will encompass personal details (names, contact information, addresses), comprehensive medical histories, detailed care plans, and essential emergency contact information.
2. **Worker Information:** We assume that worker information will include qualifications, certifications, work histories, availabilities, and contact details to ensure we have a qualified and available workforce.
3. **Scheduling and Appointments:** We assume that the systems will handle scheduling and appointment management, enabling us to efficiently assign workers to client appointments.
4. **Billing and Payments:** We assume that the systems will support billing and payment processing, including generating invoices, tracking payments, and managing accounts receivable for our financial operations.
5. **Compliance and Reporting:** We assume that we will need tools for compliance monitoring, generating compliance reports, and maintaining records to meet regulatory requirements.
6. **Security Measures:** We assume that implementing robust security measures, such as data encryption, access controls, and audit trails, is essential to protect sensitive client and worker information.
7. **Mobile Accessibility:** We assume that caregivers in the field will require mobile accessibility, either through dedicated mobile apps or mobile-friendly web interfaces.
8. **Reporting and Analytics:** We assume that our agency will benefit from reporting and analytics capabilities to assist with strategic decision-making and performance monitoring.
9. **Scalability:** We assume that the database designs must account for scalability to accommodate our agency's growth and handle increased client and worker volumes.
10. **Regulatory Compliance:** We assume that the databases will play a crucial role in helping our agency comply with relevant healthcare regulations and industry standards.

11. **Audit Trails:** We assume that we must maintain comprehensive audit trails to track and log all system activities and changes for accountability and security.
12. **User Roles and Privileges:** We assume that the systems should define user roles and access privileges based on user categories and characteristics to ensure appropriate data access and control.
13. **Training and Support:** We assume that our agency will provide training resources and technical support to help our users become proficient in using the systems effectively.
14. **Integration:** We assume that the databases may need to integrate with other systems or tools we use, such as accounting software or electronic health records (EHR) systems.
15. **Data Backup and Recovery:** We assume that regular data backup and recovery procedures will be in place to safeguard against data loss.

These assumptions provide a solid starting point for our database design projects, but we understand the importance of validating them with our stakeholders and adjusting them as needed to ensure that our databases align precisely with our agency's unique requirements and goals. Additionally, documenting these assumptions is crucial for effective project planning and risk management.

6. Business Constraints

Business constraints are factors or limitations that can affect the design, development, and operation of a Home Care Worker Providing Agency Database Management system. These constraints may arise from the agency's operations, industry regulations, budget considerations, or other factors. Here are some common business constraints that could impact the database project:

- **Budget Constraints:**
The agency may have a limited budget for the development, implementation, and maintenance of the database system. This budget constraint can influence the choice of technology, features, and the scope of the project.
- **Timeline Constraints:**
There may be a specific timeline or deadline for implementing the system. For example, the agency might need the system up and running within a certain timeframe to meet regulatory requirements or operational needs.
- **Regulatory Compliance:**

Healthcare agencies are subject to various regulations and compliance standards (e.g., HIPAA in the United States). These regulations impose constraints on data security, privacy, and reporting, which must be strictly adhered to in the system design.

- **Data Privacy and Security:**

Protecting sensitive client and worker information is a paramount concern. Business constraints may dictate the need for stringent data security measures, encryption, and access controls, which can impact the system's design.

- **Integration Requirements:**

The agency may already have existing software systems or tools (e.g., electronic health records, accounting software) that need to be integrated with the new database system. Integration constraints can influence the choice of technology and architecture.

- **User Training and Adoption:**

User adoption is crucial for the system's success. Constraints related to user training and change management may require additional resources and planning.

- **Scalability:**

The agency's growth projections and scalability requirements can be a constraint. The system must be designed to accommodate an increasing number of clients and workers without major disruptions.

- **Resource Availability:**

Availability of skilled IT staff, developers, and support personnel can be a constraint. The agency may need to rely on external resources or contractors, which can impact project costs and timelines.

- **Client and Worker Data Volume:**

The sheer volume of client and worker data can be a constraint, especially in larger agencies. The system must efficiently handle large datasets while maintaining performance.

- **Operational Continuity:**

Ensuring uninterrupted home care services is a critical constraint. The system should have backup and recovery mechanisms in place to prevent service disruptions.

- **Change Management:**

Introducing a new system can lead to resistance from staff. Business constraints may necessitate comprehensive change management strategies to ensure smooth adoption.

- **Reporting and Analytics:**

Specific reporting requirements imposed by regulatory bodies or funding sources can be a constraint. The system must generate and submit required reports accurately and on time.

- **Billing and Payment Processing:**

Business constraints related to billing and payment processing, such as reimbursement policies or insurance requirements, can affect the system's billing module.

- **Geographic Constraints:**

The geographic location and service area of the agency can influence system requirements, especially if the agency operates in multiple regions with varying regulations.

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

Serial Number	Noun	Verb
1	Home Care Worker	Represents
2	Agency	Serves
3	Database	Alignment
4	Management system	Ensuring
5	Tool	Integration
6	Backbone	Designed
7	Home care workers	Facilitating
8	Clients	Empowers
9	Services	Gain
10	Administrative processes	Facilitate
11	Strategic objectives	Operates
12	Mission	Promotes
13	High-quality home-based care services	Access
14	Operational efficiency	Generate
15	Regulatory standards	Improve
16	Trends	Streamlines
17	Healthcare	Automates
18	Telehealth integration	Optimizes
19	Mobile care delivery	Providing
20	Data-driven decision-making	Built
21	Client-centred care	Encompasses
22	Comprehensive medical records	Coordinates
23	Compliance	Manages

Serial Number	Noun	Verb
24	Healthcare regulations	Tailor
25	Industry standards	Maintains
26	Data	Efficiently match
27	Decision-making	Facilitates
28	Managers	Accommodate
29	Reports	Adapts
30	Analytics	Protect
31	Stakeholders	Evolves
32	User experience	Updates
33	Staff	Measure
34	Client satisfaction	Aims
35	Outcomes	Enhance
36	Software solution	Ensures
37	Client data	Handles
38	Personal information	Safeguard
39	Medical history	Support
40	Care requirements	Catering
41	Contact details	Designed
42	Home Care Worker Profiles	
43	Qualifications	
44	Certifications	
45	Work history	
46	Availability	
47	Scheduling module	

Serial Number	Noun	Verb
48	Matching algorithms	
49	Recurring visits	
50	Invoices	
51	Payment processing	
52	Payroll	
53	Compliance records	
54	Documentation	
55	Performance monitoring	
56	Data security	
57	Privacy measures	
58	Billing	
59	Payment processes	
60	Scheduling	
61	Technical backgrounds	
62	Operating systems	
63	Web browsers	
64	Mobile devices	
65	Continuous improvement	
66	User feedback	
67	Client experience	
68	Health	
69	Viability	
70	Disaster recovery	
71	Server specifications	

Serial Number	Noun	Verb
72	Client devices	
73	Third-party software	
74	Security measures	
75	Backup	
76	Client base	
77	Technical expertise	
78	Mobile application	
79	Configurability	
80	Scalable architecture	

Table.1. Candidate Entity Set

Home Care Worker	Represents a primary entity in the system responsible for providing care.
Agency	Represents the organization that manages home care services.
Client	Represents individuals who receive home care services.
Service	Represents the different types of services provided to clients.
Qualification	Represents the qualifications held by home care workers.
Certification	Represents certifications obtained by home care workers.
Work History	Represents the work history of home care workers.
Schedule	Represents the scheduling of appointments for clients and home care workers.
Visit	Represents individual home care visits or appointments.
Invoice	Represents billing information for services provided to clients.
Payment	Represents payments received from clients.
Compliance Record	Represents records of compliance with healthcare regulations.

Medical Record	Represents comprehensive medical records for clients.
Report	Represents generated reports and analytics.
Analytics	Represents data analysis and analytics.
User	Represents users of the system.
Third-Party Software	Represents external software components used in the system.

Table.2. Candidate Attributes Set

Noun	Attributes
Home Care Worker	Name, Contact Information
Agency	Name, Address
Client	Name, Date of Birth
Service	Service Name, Service Description
Qualification	Qualification Name, Date of Certification
Certification	Certification Type, Expiry Date
Work History	Previous Employers, Job Titles and Responsibilities
Schedule	Date and Time, Location
Visit	Visit Date, Visit Type
Invoice	Invoice Number, Total Amount
Payment	Payment Date, Payment Method
Compliance Record	Compliance Date, Regulatory Requirements Met
Medical Record	Medical History, Diagnosis and Treatment Records
Report	Report Name, Report Date
Analytics	Data Source, Analysis Results
User	Username, User Role
Third-Party Software	Software Name, Vendor Information

Table.3. Rejected Noun List

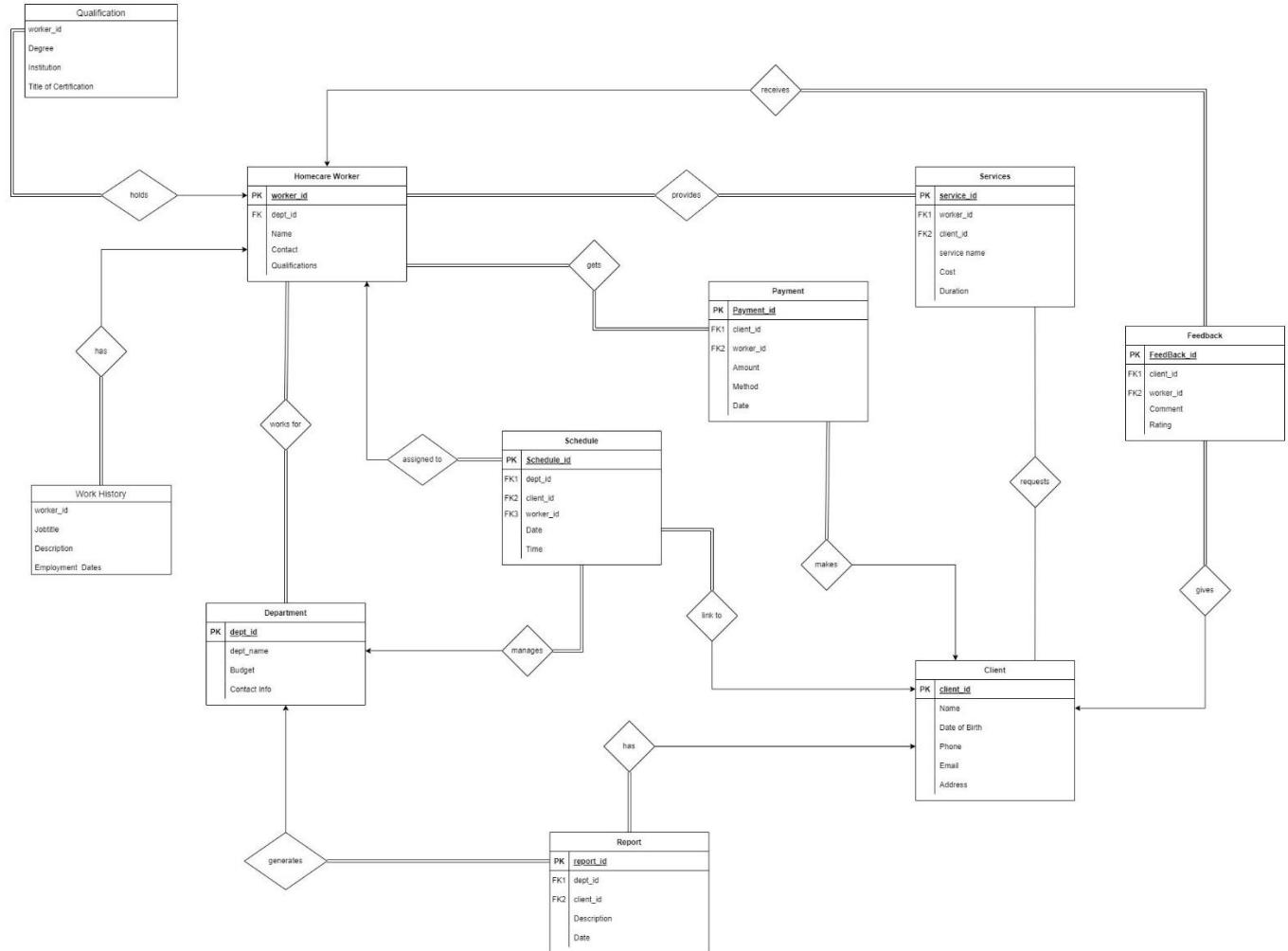
Noun	Reject Reason
Telehealth Integration	Not part of the intended system
Mobile Care Delivery	Not part of the intended system
Data-Driven	General (lacks specificity)
Catalyst	General (lacks specificity)
Viability	General (lacks specificity)
Long-Term	General (lacks specificity)
Market Trends	Not part of the intended system
Competition	Not part of the intended system
Innovation	Not part of the intended system
Customer Satisfaction	Not part of the intended system
Profitability	Not part of the intended system
Corporate Strategy	Not part of the intended system
Leadership	Not part of the intended system
Vision	Not part of the intended system
Mission	Not part of the intended system
Trend Analysis	Not part of the intended system
Forecasting	Not part of the intended system
Best Practices	Not part of the intended system
Quality Improvement	Not part of the intended system
Strategic Planning	Not part of the intended system
Stakeholders	Not part of the intended system
SWOT Analysis	Not part of the intended system
ROI	Not part of the intended system

Noun	Reject Reason
Marketing Strategy	Not part of the intended system
Branding	Not part of the intended system
Digital Marketing	Not part of the intended system
Advertising	Not part of the intended system
Social Media	Not part of the intended system
Customer Engagement	Not part of the intended system

Table.1.1. Final Entity Set

Noun	Attributes
Home Care Worker	Worker_ID , Name, Contact, Qualification
Client	Client_ID , Name, Date Of Birth, Phone, Email, Address
Service	Service_ID , ServiceName, Duration, Cost
Qualification	Institution, Degree, Title of certification
Work History	Employment Dates, JobTitle, Description
Schedule	Schedule_ID, Date, Time
Payment	Payment_ID , Amount, Date, Method
Department	Dept_ID , Dept_Name, Budget, Contact info
Report	Report_ID , ReportDate, Description
Feedback	FB_ID , Comment, Rating

ENTITY RELATIONSHIP DIAGRAM



Relational Schema

Client (Client_ID , Name, Date Of Birth, Phone, Email, Address)

Home Care Worker(Worker_ID , Name, Contact, Qualification, Dept_ID (FK))

Service(Service_ID , ServiceName, Duration, Cost, Worker_ID(FK), Client_ID(FK))

Schedule(Schedule_ID, Date, Time, Client_ID(FK), Worker_ID(FK), Dept_ID (FK))

Payment(Payment_ID , Amount, Date, Method, Client_ID(FK), Worker_ID (FK))

Qualification(Institution, Degree, Title of certification, Worker_ID(FK))

Work History(Employment Dates, JobTitle, Description, WorkerID(FK))

Department(Dept_ID , Dept_Name, Budget, Contact info)

Report(Report_ID , ReportDate, Description, Client_ID (FK), Dept_ID (FK))

Feedback(FB_ID , Comment, Rating, Worker_ID (FK), Client_ID (FK))

List of Redundancies for Each Schema:

Client Table:

- Redundancy: Contact information (Phone, Email) could be duplicated if multiple clients share the same contact details. Normalization could involve separating contact information into a separate table.

Home Care Worker Table:

- Redundancy: Contact details might be redundant if multiple workers share the same contact information. Normalizing this information could involve creating a separate table for contacts.

Service Table:

- Redundancy: Worker_ID and Client_ID could potentially duplicate information present in the Home Care Worker and Client tables. Repeated use of these foreign keys might lead to data inconsistency if not managed properly.

Schedule Table:

- Redundancy: Date and Time information could be standardized if multiple schedules share the same date and time. A separate table could manage standardized date-time entries.

Payment Table:

- Redundancy: Payment details like Amount, Date, and Method might be repeated if multiple payments are made by the same client or worker. Normalization could involve Payment Method and Date.

Qualification Table:

- Redundancy: Qualification details might repeat for various workers. Normalization could involve creating a separate table for qualifications to avoid duplicating Institution, Degree, and Title of Certification information.

Work History Table:

- Redundancy: Employment Dates, Job Title, and Description might recur for multiple workers. Normalization could involve structuring the data in a way that minimizes repetition of this information.

Department Table:

- Redundancy: Contact information might be duplicated if multiple departments share the same contact details. Normalization could involve creating a separate table for contact details.

Report Table:

- Redundancy: ReportDate and Description might repeat for different reports. Normalization could involve standardizing descriptions or creating separate tables to minimize this repetition.

Feedback Table:

- Redundancy: Comment and Rating might repeat for various feedback entries. Normalization could involve structuring the data to minimize repetition of comments and ratings.

List of Update, Delete, and Insert Anomalies for Each Schema:

Client Table:

- Update Anomaly: If a client's contact information changes, updating it in multiple places (if duplicated) could lead to inconsistencies.
- Delete Anomaly: Deleting a client might result in losing associated contact information if not appropriately managed.
- Insert Anomaly: Inability to insert a client without contact information if it's in a separate table.

Home Care Worker Table:

- Update Anomaly: Updating contact details might require changes in multiple places if duplicated.
- Delete Anomaly: Deleting a worker could result in losing associated contact information if not handled correctly.
- Insert Anomaly: Difficulty in adding a worker without contact information if it's in a separate table.

Service Table:

- Update Anomaly: Altering worker or client information in multiple rows can be error-prone.
- Delete Anomaly: Removing a service might unintentionally delete associated worker or client details.
- Insert Anomaly: Inability to add a service without an associated worker or client.

Schedule Table:

- Update Anomaly: Changing worker or client details in multiple rows might lead to inconsistencies.
- Delete Anomaly: Deleting a schedule might unintentionally delete worker or client information if not managed properly.
- Insert Anomaly: Difficulty in adding a schedule without associated worker or client information.

Payment Table:

- Update Anomaly: Modifying payment details across multiple rows might lead to inconsistencies.
- Delete Anomaly: Deleting a payment might inadvertently remove related client or worker data.
- Insert Anomaly: Inability to insert a payment without associated client or worker information.

Qualification Table:

- Update Anomaly: Changing qualification details across multiple rows might lead to inconsistencies.
- Delete Anomaly: Deleting a qualification might unintentionally remove associated worker data.
- Insert Anomaly: Difficulty in adding qualifications without worker information.

Work History Table:

- Update Anomaly: Modifying work history across multiple rows might lead to inconsistencies.
- Delete Anomaly: Deleting work history might unintentionally remove associated worker data.
- Insert Anomaly: Inability to add work history without worker information.

Department Table:

- Update Anomaly: Changing department details across multiple rows might lead to inconsistencies.
- Delete Anomaly: Deleting a department might unintentionally remove associated information.
- Insert Anomaly: Difficulty in adding a department without associated information.

Report Table:

- Update Anomaly: Modifying report details across multiple rows might lead to inconsistencies.
- Delete Anomaly: Deleting a report might unintentionally remove related client or department data.
- Insert Anomaly: Inability to insert a report without associated client or department information.

Feedback Table:

- Update Anomaly: Changing feedback details across multiple rows might lead to inconsistencies.
- Delete Anomaly: Deleting feedback might inadvertently remove related client or worker data.
- Insert Anomaly: Difficulty in adding feedback without associated client or worker information.

Normalization and Schema

Step-by-Step Schema Refinement for the "Client" Table

Step 1: Achieving First Normal Form (1NF)

Our initial "Client" table is as follows:

Client (Client_ID PK, Name, Date Of Birth, Phone, Email, Address)

1.1 Remove Repeating Groups

In our original table, there are no repeating groups, so it already satisfies 1NF.

1.2 Atomic Values

All attributes in the table contain atomic (indivisible) values, so we meet this criterion for 1NF.

Our "Client" table in 1NF remains unchanged:

Client (Client_ID PK, Name, Date Of Birth, Phone, Email, Address)

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies

To move to 2NF, we must determine if there are any partial dependencies.

- Candidate key: Client_ID (as it uniquely identifies each client)

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

Thus, our "Client" table remains in 2NF:

Client (Client_ID PK, Name, Date Of Birth, Phone, Email, Address)

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies

To reach 3NF, we need to address transitive dependencies.

- Candidate key: Client_ID
- We observe that Phone and Email are functionally dependent on each other. This means they are transitively dependent.

To resolve this, we'll create a new table called "ContactInfo" with a unique identifier (Contact_ID) and move Phone and Email into this table. The "Client" table will reference the "ContactInfo" table using Contact_ID. This eliminates the transitive dependency.

Our "Client" table in 3NF is as follows:

Client (Client_ID PK, Name, Date Of Birth, Contact_ID (FK), Address)

ContactInfo (Contact_ID PK, Phone, Email)

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

4.1 Remove Partial Dependencies

In the 3NF version, there are no partial dependencies.

4.2 Check for Candidate Keys

The candidate key in the original table is Client_ID, and it remains the candidate key in our refined table.

4.3 Check for Non-Prime Attributes Dependencies

There are no non-prime attributes dependent on any proper subset of the candidate key, ensuring that our table is in BCNF.

In conclusion, our refined "Client" table satisfies 1NF, 2NF, 3NF, and BCNF. This schema eliminates data redundancy and anomalies, ensuring data integrity and efficiency.

Step-by-Step Schema Refinement for the "Home Care Worker" Table

Step 1: Achieving First Normal Form (1NF)

Our initial "Home Care Worker" table is as follows:

Home Care Worker (Worker_ID PK, Name, Contact, Qualification, Dept_ID (FK))

1.1 Remove Repeating Groups

In our original table, there are no repeating groups, so it already satisfies 1NF.

1.2 Atomic Values

All attributes in the table contain atomic (indivisible) values, so we meet this criterion for 1NF.

Our "Home Care Worker" table in 1NF remains unchanged:

Home Care Worker (Worker_ID PK, Name, Contact, Qualification, Dept_ID (FK))

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies

To move to 2NF, we must determine if there are any partial dependencies.

- Candidate key: Worker_ID (as it uniquely identifies each worker)

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

Thus, our "Home Care Worker" table remains in 2NF:

Home Care Worker (Worker_ID PK, Name, Contact, Qualification, Dept_ID (FK))

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies

To reach 3NF, we need to address transitive dependencies.

- Candidate key: Worker_ID
- The attributes Name, Contact, and Qualification do not transitively depend on any other non-key attributes. They depend directly on the candidate key.
- However, the "Contact" attribute combines both the worker's phone and email. This is a transitive dependency.

To eliminate this transitive dependency, we will create a new table for "ContactInfo" with a unique identifier (Contact_ID) and move the "Contact" attribute into this table. The "Home Care Worker" table will reference the "ContactInfo" table using Contact_ID.

Our "Home Care Worker" table in 3NF is as follows:

Home Care Worker (Worker_ID PK, Name, Contact_ID (FK), Qualification, Dept_ID (FK))

ContactInfo (Contact_ID PK, Phone, Email)

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

4.1 Remove Partial Dependencies

In the 3NF version, there are no partial dependencies.

4.2 Check for Candidate Keys

The candidate key in the original table is Worker_ID, and it remains the candidate key in our refined table.

4.3 Check for Non-Prime Attributes Dependencies

There are no non-prime attributes dependent on any proper subset of the candidate key, ensuring that our table is in BCNF.

In conclusion, our refined "Home Care Worker" table satisfies 1NF, 2NF, 3NF, and BCNF. This schema eliminates data redundancy and anomalies, ensuring data integrity and efficiency.

Step-by-Step Schema Refinement for the "Service" Table

Step 1: Achieving First Normal Form (1NF)

Our initial "Service" table is as follows:

Service (Service_ID PK, ServiceName, Duration, Cost, Worker_ID(FK), Client_ID(FK))

1.1 Remove Repeating Groups

In our original table, there are no repeating groups, so it already satisfies 1NF.

1.2 Atomic Values

All attributes in the table contain atomic (indivisible) values, so we meet this criterion for 1NF.

Our "Service" table in 1NF remains unchanged:

Service (Service_ID PK, ServiceName, Duration, Cost, Worker_ID(FK), Client_ID(FK))

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies

To move to 2NF, we must determine if there are any partial dependencies.

Candidate key: {Service_ID, Worker_ID, Client_ID}

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

Thus, our "Service" table remains in 2NF:

Service (Service_ID PK, ServiceName, Duration, Cost, Worker_ID(FK), Client_ID(FK))

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies

To reach 3NF, we need to address transitive dependencies.

- Candidate key: {Service_ID, Worker_ID, Client_ID}
- The attributes ServiceName, Duration, and Cost do not transitively depend on any other non-key attributes. They depend directly on the candidate key.

Our "Service" table in 3NF remains unchanged:

Service (Service_ID PK, ServiceName, Duration, Cost, Worker_ID(FK), Client_ID(FK))

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

4.1 Remove Partial Dependencies

In the 3NF version, there are no partial dependencies.

4.2 Check for Candidate Keys

The candidate key in the original table is {Service_ID, Worker_ID, Client_ID}, and it remains the candidate key in our refined table.

4.3 Check for Non-Prime Attributes Dependencies

There are no non-prime attributes dependent on any proper subset of the candidate key, ensuring that our table is in BCNF.

In conclusion, our refined "Service" table satisfies 1NF, 2NF, 3NF, and BCNF. This schema eliminates data redundancy and anomalies, ensuring data integrity and efficiency.

Step-by-Step Schema Refinement for the "Schedule" Table

Step 1: Achieving First Normal Form (1NF)

Our initial "Schedule" table is as follows:

Schedule (Schedule_ID PK, Date, Time, Client_ID(FK), Worker_ID(FK), Dept_ID (FK))

1.1 Remove Repeating Groups

In our original table, there are no repeating groups, so it already satisfies 1NF.

1.2 Atomic Values

All attributes in the table contain atomic (indivisible) values, so we meet this criterion for 1NF.

Our "Schedule" table in 1NF remains unchanged:

Schedule (Schedule_ID PK, Date, Time, Client_ID(FK), Worker_ID(FK), Dept_ID(FK))

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies

To move to 2NF, we must determine if there are any partial dependencies.

- Candidate key: Schedule_ID (as it uniquely identifies each schedule entry)

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

Thus, our "Schedule" table remains in 2NF:

Schedule (Schedule_ID PK, Date, Time, Client_ID(FK), Worker_ID(FK), Dept_ID(FK))

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies

To reach 3NF, we need to address transitive dependencies.

- Candidate key: Schedule_ID
- The attributes Date and Time do not transitively depend on any other non-key attributes. They depend directly on the candidate key.

Our "Schedule" table in 3NF remains unchanged:

Schedule (Schedule_ID PK, Date, Time, Client_ID(FK), Worker_ID(FK), Dept_ID(FK))

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

4.1 Remove Partial Dependencies

In the 3NF version, there are no partial dependencies.

4.2 Check for Candidate Keys

The candidate key in the original table is Schedule_ID, and it remains the candidate key in our refined table.

4.3 Check for Non-Prime Attributes Dependencies

There are no non-prime attributes dependent on any proper subset of the candidate key, ensuring that our table is in BCNF.

In conclusion, our refined "Schedule" table satisfies 1NF, 2NF, 3NF, and BCNF. This schema eliminates data redundancy and anomalies, ensuring data integrity and efficiency.

Step-by-Step Schema Refinement for the "Payment" Table

Step 1: Achieving First Normal Form (1NF)

Our initial "Payment" table is as follows:

Payment (Payment_ID, Amount, Date, Method, Client_ID(FK), Worker_ID(FK))

1.1 Remove Repeating Groups

In our original table, there are no repeating groups, so it already satisfies 1NF.

1.2 Atomic Values

All attributes in the table contain atomic (indivisible) values, so we meet this criterion for 1NF.

Our "Payment" table in 1NF remains unchanged:

Payment (Payment_ID, Amount, Date, Method, Client_ID(FK), Worker_ID(FK))

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies

To move to 2NF, we must determine if there are any partial dependencies.

- Candidate key: {Payment_ID, Client_ID, Worker_ID} (as it uniquely identifies each payment)

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

Thus, our "Payment" table remains in 2NF:

Payment (Payment_ID, Amount, Date, Method, Client_ID(FK), Worker_ID(FK))

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies

To reach 3NF, we need to address transitive dependencies.

- Candidate key: {Payment_ID, Client_ID, Worker_ID}
- The attributes Amount, Date, and Method do not transitively depend on any other non-key attributes. They depend directly on the candidate key.

Our "Payment" table in 3NF remains unchanged:

Payment (Payment_ID, Amount, Date, Method, Client_ID(FK), Worker_ID(FK))

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

4.1 Remove Partial Dependencies

In the 3NF version, there are no partial dependencies.

4.2 Check for Candidate Keys

The candidate key in the original table is {Payment_ID, Client_ID, Worker_ID}, and it remains the candidate key in our refined table.

4.3 Check for Non-Prime Attributes Dependencies

There are no non-prime attributes dependent on any proper subset of the candidate key, ensuring that our table is in BCNF.

In conclusion, our refined "Payment" table satisfies 1NF, 2NF, 3NF, and BCNF. This schema eliminates data redundancy and anomalies, ensuring data integrity and efficiency.

Step-by-Step Schema Refinement for the "Qualification" Table

Step 1: Achieving First Normal Form (1NF)

The initial "Qualification" table looks like this:

Qualification(Institution, Degree, Title of certification, Worker_ID(FK))

1.1 Remove Repeating Groups The table doesn't contain repeating groups, so it already satisfies 1NF.

1.2 Atomic Values All attributes seem to contain atomic values.

The "Qualification" table remains in 1NF:

Qualification(Institution, Degree, Title of certification, Worker_ID(FK))

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies Identify the candidate key:

- Candidate keys: {Worker_ID, Institution, Degree, Title of certification}

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

The "Qualification" table remains in 2NF:

Qualification(Institution, Degree, Title of certification, Worker_ID(FK))

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies The table doesn't show transitive dependencies.

The "Qualification" table remains the same:

Qualification(Institution, Degree, Title of certification, Worker_ID(FK))

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

The "Qualification" table doesn't contain any partial dependencies or non-prime attribute dependencies. It already satisfies BCNF.

In conclusion, the "Qualification" table appears to be well-structured without redundancies or data anomalies, meeting the criteria of 1NF, 2NF, 3NF, and BCNF.

Step-by-Step Schema Refinement for the "Work History" Table

Step 1: Achieving First Normal Form (1NF)

The initial "Work History" table looks like this:

sqlCopy code

```
Work History(Employment Dates, JobTitle, Description, Worker_ID(FK))
```

1.1 Remove Repeating Groups The table doesn't contain repeating groups, so it already satisfies 1NF.

1.2 Atomic Values All attributes seem to contain atomic values.

The "Work History" table remains in 1NF:

sqlCopy code

```
Work History(Employment Dates, JobTitle, Description, Worker_ID(FK))
```

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies Identify the candidate key:

- Candidate keys: {Worker_ID, Employment Dates, JobTitle}

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

The "Work History" table remains in 2NF:

sqlCopy code

```
Work History(Employment Dates, JobTitle, Description, Worker_ID(FK))
```

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies The table doesn't show transitive dependencies.

The "Work History" table remains the same:

sqlCopy code

```
Work History(Employment Dates, JobTitle, Description, Worker_ID(FK))
```

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

The "Work History" table doesn't contain any partial dependencies or non-prime attribute dependencies. It already satisfies BCNF.

In conclusion, the "Work History" table appears to be well-structured without redundancies or data anomalies, meeting the criteria of 1NF, 2NF, 3NF, and BCNF.

Step-by-Step Schema Refinement for the "Department" Table

Step 1: Achieving First Normal Form (1NF)

Our initial "Department" table is as follows:

Department (Dept_ID PK, Dept_Name, Budget, Contact info)

1.1 Remove Repeating Groups

In our original table, there are no repeating groups, so it already satisfies 1NF.

1.2 Atomic Values

All attributes in the table contain atomic (indivisible) values, so we meet this criterion for 1NF.

Department (Dept_ID PK, Dept_Name, Budget , Contact info)

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies

To move to 2NF, we must determine if there are any partial dependencies.

- Candidate key: Dept_ID (as it uniquely identifies each department)

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

Thus, our "Department" table remains in 2NF:

Department (Dept_ID PK, Dept_Name, Budget, Contact info)

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies

To reach 3NF, we need to address transitive dependencies.

- Candidate key: Dept_ID
- The attributes Dept_Name and Address do not transitively depend on any other non-key attributes. They depend directly on the candidate key.

Our "Department" table in 3NF remains unchanged:

Department (Dept_ID PK, Dept_Name, Budget, Contact info)

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

4.1 Remove Partial Dependencies

In the 3NF version, there are no partial dependencies.

4.2 Check for Candidate Keys

The candidate key in the original table is Dept_ID, and it remains the candidate key in our refined table.

4.3 Check for Non-Prime Attributes Dependencies

There are no non-prime attributes dependent on any proper subset of the candidate key, ensuring that our table is in BCNF.

In conclusion, our refined "Department" table satisfies 1NF, 2NF, 3NF, and BCNF. This schema eliminates data redundancy and anomalies, ensuring data integrity and efficiency.

Step-by-Step Schema Refinement for the "Report" Table

Step 1: Achieving First Normal Form (1NF)

Our initial "Report" table is as follows:

Report (Report_ID, ReportDate, Description, Client_ID(FK), Dept_ID(FK))

1.1 Remove Repeating Groups

In our original table, there are no repeating groups, so it already satisfies 1NF.

1.2 Atomic Values

All attributes in the table contain atomic (indivisible) values, so we meet this criterion for 1NF.

Our "Report" table in 1NF remains unchanged:

Report (Report_ID, ReportDate, Description, Client_ID(FK), Dept_ID(FK))

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies

To move to 2NF, we must determine if there are any partial dependencies.

- Candidate key: {Report_ID, Client_ID, Dept_ID} (as it uniquely identifies each report)

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

Thus, our "Report" table remains in 2NF:

Report (Report_ID, ReportDate, Description, Client_ID(FK), Dept_ID(FK))

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies

To reach 3NF, we need to address transitive dependencies.

- Candidate key: {Report_ID, Client_ID, Dept_ID}
- The attributes ReportDate and Description do not transitively depend on any other non-key attributes. They depend directly on the candidate key.

Our "Report" table in 3NF remains unchanged:

Report (Report_ID, ReportDate, Description, Client_ID(FK), Dept_ID(FK))

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

4.1 Remove Partial Dependencies

In the 3NF version, there are no partial dependencies.

4.2 Check for Candidate Keys

The candidate key in the original table is {Report_ID, Client_ID, Dept_ID}, and it remains the candidate key in our refined table.

4.3 Check for Non-Prime Attributes Dependencies

There are no non-prime attributes dependent on any proper subset of the candidate key, ensuring that our table is in BCNF.

In conclusion, our refined "Report" table satisfies 1NF, 2NF, 3NF, and BCNF. This schema eliminates data redundancy and anomalies, ensuring data integrity and efficiency.

Step-by-Step Schema Refinement for the "Feedback" Table

Step 1: Achieving First Normal Form (1NF)

Our initial "Feedback" table is as follows:

Feedback (FB_ID, Comment, Rating, Worker_ID(FK), Client_ID(FK))

1.1 Remove Repeating Groups

In our original table, there are no repeating groups, so it already satisfies 1NF.

1.2 Atomic Values

All attributes in the table contain atomic (indivisible) values, so we meet this criterion for 1NF.

Our "Feedback" table in 1NF remains unchanged:

Feedback (FB_ID, Comment, Rating, Worker_ID(FK), Client_ID(FK))

Step 2: Achieving Second Normal Form (2NF)

2.1 Remove Partial Dependencies

To move to 2NF, we must determine if there are any partial dependencies.

- Candidate key: {FB_ID, Worker_ID, Client_ID} (as it uniquely identifies each feedback record)

In our current schema, there are no partial dependencies. All non-key attributes are fully functionally dependent on the entire candidate key.

Thus, our "Feedback" table remains in 2NF:

Feedback (FB_ID, Comment, Rating, Worker_ID(FK), Client_ID(FK))

Step 3: Achieving Third Normal Form (3NF)

3.1 Remove Transitive Dependencies

To reach 3NF, we need to address transitive dependencies.

- Candidate key: {FB_ID, Worker_ID, Client_ID}
- The attributes Comment and Rating do not transitively depend on any other non-key attributes. They depend directly on the candidate key.

Our "Feedback" table in 3NF remains unchanged:

Feedback (FB_ID, Comment, Rating, Worker_ID(FK), Client_ID(FK))

Step 4: Achieving Boyce-Codd Normal Form (BCNF)

4.1 Remove Partial Dependencies

In the 3NF version, there are no partial dependencies.

4.2 Check for Candidate Keys

The candidate key in the original table is {FB_ID, Worker_ID, Client_ID}, and it remains the candidate key in our refined table.

4.3 Check for Non-Prime Attributes Dependencies

There are no non-prime attributes dependent on any proper subset of the candidate key, ensuring that our table is in BCNF.

In conclusion, our refined "Feedback" table satisfies 1NF, 2NF, 3NF, and BCNF. This schema eliminates data redundancy and anomalies, ensuring data integrity and efficiency.

Final Relation Schema

Client Table:

Client (Client_ID , Name, Date Of Birth, Contact_ID (FK), Address)

ContactInfo (Contact_ID , Phone, Email)

Home Care Worker Table:

Home Care Worker (Worker_ID , Name, Qualification, Dept_ID (FK))

Service Table:

Service (Service_ID , ServiceName, Duration, Cost, Worker_ID (FK), Client_ID (FK))

Schedule Table:

Schedule (Schedule_ID , Date, Time, Client_ID (FK), Worker_ID (FK), Dept_ID (FK))

Payment Table:

Payment (Payment_ID , Amount, Date, Method, Client_ID (FK), Worker_ID (FK))

Qualification Table:

Qualification (Institution, Degree, Title of certification, Worker_ID (FK))

Work History Table:

Work History (Employment Dates, JobTitle, Description, WorkerID (FK))

Department Table:

Department (Dept_ID , Dept_Name, Budget, Contact info)

Report Table:

Report (Report_ID , ReportDate, Description, Client_ID (FK), Dept_ID (FK))

Feedback Table:

Feedback (FB_ID , Comment, Rating, Worker_ID (FK), Client_ID (FK))

DDL Script and Snapshots

-- Client Table

```
CREATE TABLE Client (
```

```
    Client_ID SERIAL PRIMARY KEY,
```

```
    Name VARCHAR(100) NOT NULL,
```

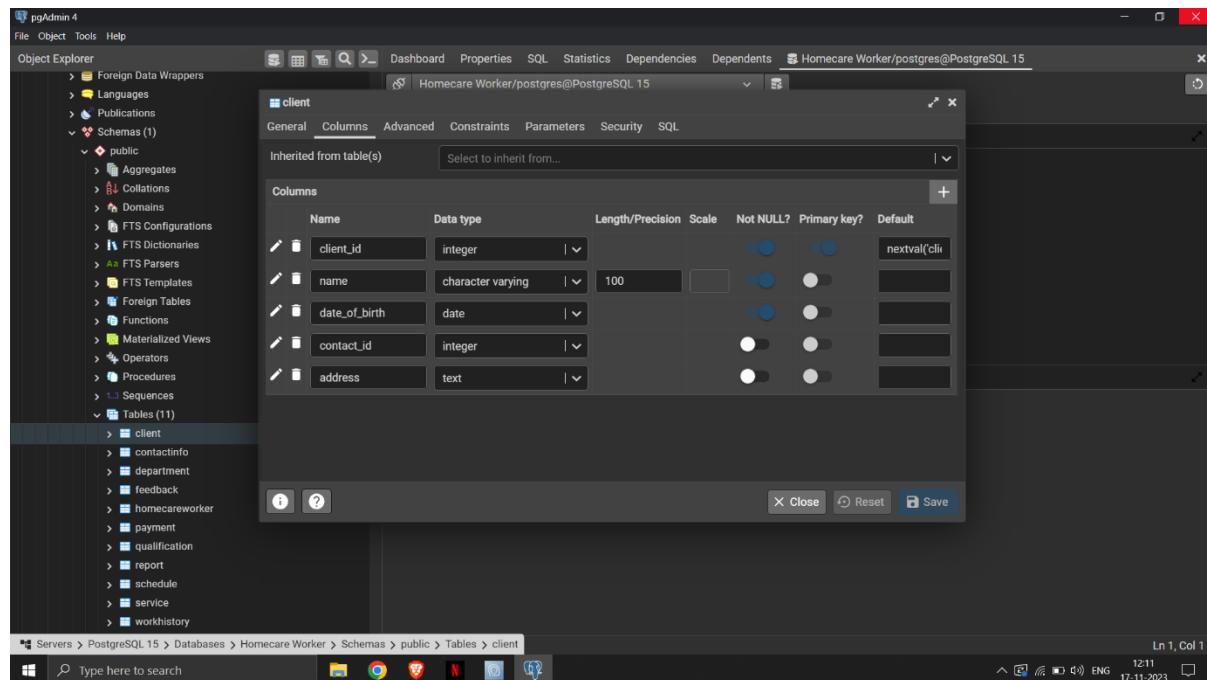
```
    Date_Of_Birth DATE NOT NULL,
```

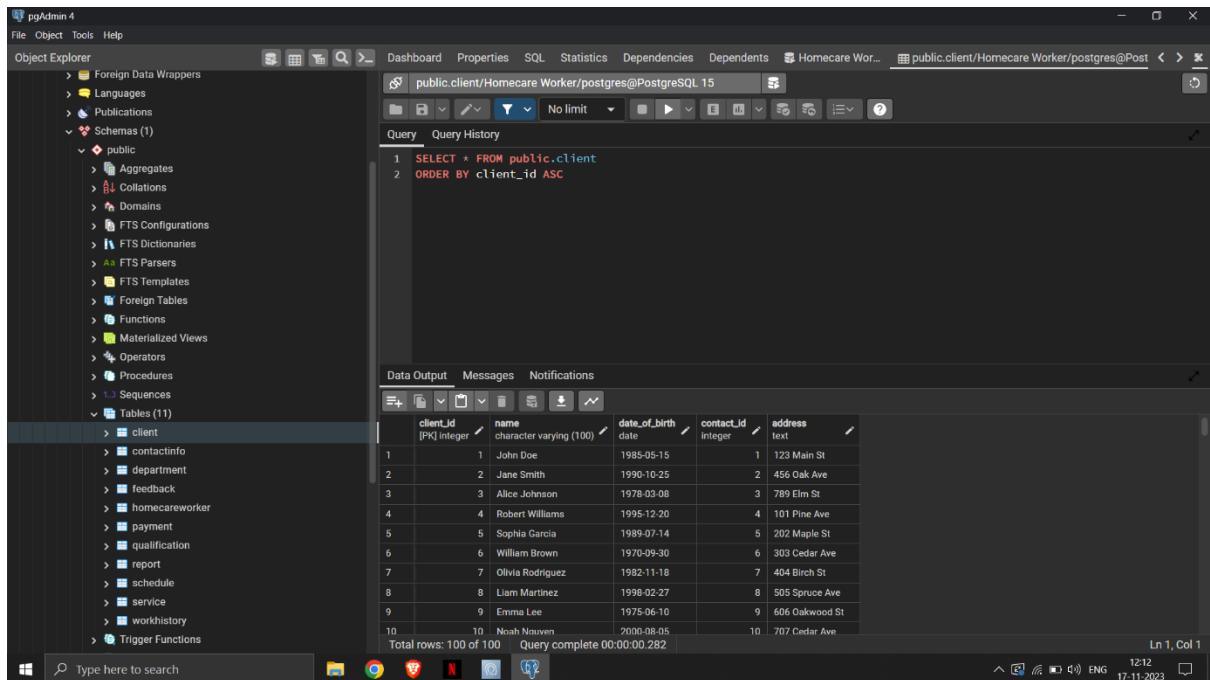
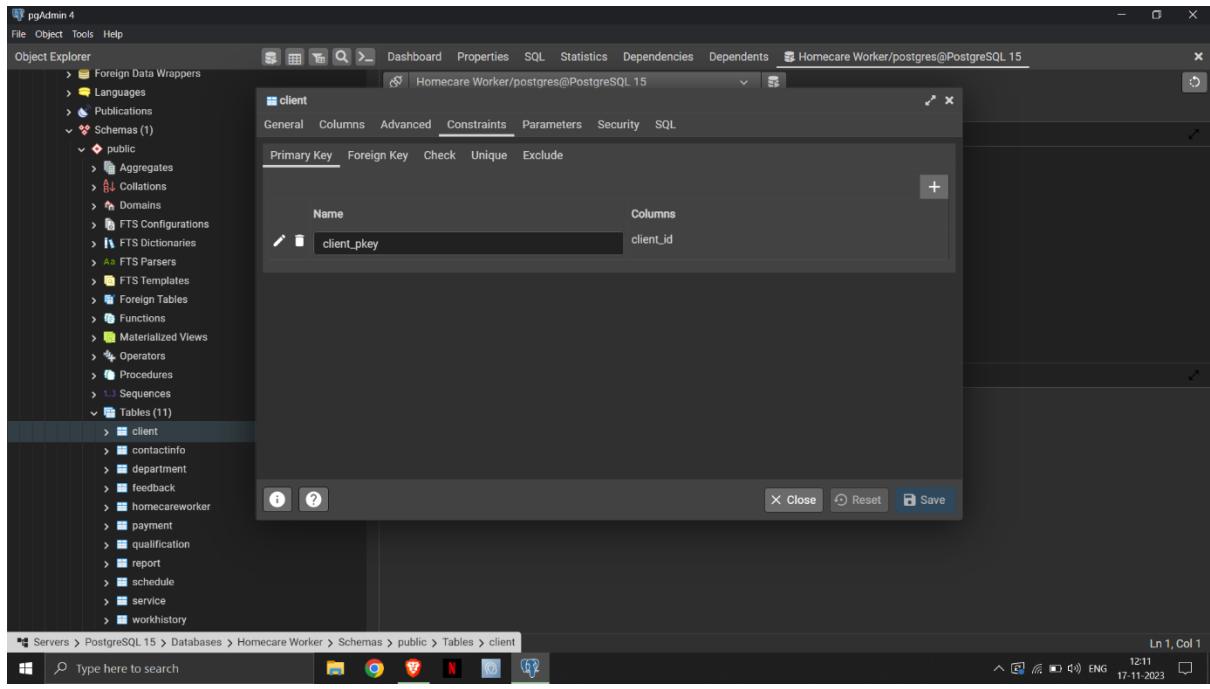
```
    Contact_ID INTEGER,
```

```
    Address TEXT,
```

```
    CONSTRAINT client_date_of_birth CHECK (Date_Of_Birth <= CURRENT_DATE)
```

```
);
```





-- ContactInfo Table

CREATE TABLE ContactInfo (

Contact_ID SERIAL PRIMARY KEY,

Phone VARCHAR(20),

Email VARCHAR(100)

);

pgAdmin 4

File Object Tools Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents public.client/Homecare Worker/postgres@PostgreSQL 15

contactinfo

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
contact_id	integer			Yes	No	nextval('co...')
phone	character varying	20		No	No	
email	character varying	100		No	No	

Save

5	5	Sophia Garcia	1989-07-14	5	202 Maple St
6	6	William Brown	1970-09-30	6	303 Cedar Ave
7	7	Olivia Rodriguez	1982-11-18	7	404 Birch St
8	8	Liam Martinez	1998-02-27	8	505 Spruce Ave
9	9	Emma Lee	1975-06-10	9	606 Oakwood St
10	10	Noah Nauven	2000-08-05	10	707 Cedar Ave

Total rows: 100 of 100 Query complete 00:00:00.282

Ln 1, Col 1

Type here to search

12:12 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents public.client/Homecare Worker/postgres@PostgreSQL 15

contactinfo

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns
contactinfo_pkey	contact_id

Save

5	5	Sophia Garcia	1989-07-14	5	202 Maple St
6	6	William Brown	1970-09-30	6	303 Cedar Ave
7	7	Olivia Rodriguez	1982-11-18	7	404 Birch St
8	8	Liam Martinez	1998-02-27	8	505 Spruce Ave
9	9	Emma Lee	1975-06-10	9	606 Oakwood St
10	10	Noah Nauven	2000-08-05	10	707 Cedar Ave

Total rows: 100 of 100 Query complete 00:00:00.282

Ln 1, Col 1

Type here to search

12:13 17-11-2023

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the 'public' schema, there are 11 tables listed: client, contactinfo, department, feedback, homecareworker, payment, qualification, report, schedule, service, and workhistory. The 'contactinfo' table is currently selected. In the main pane, a SQL query is run:

```

1 SELECT * FROM public.contactinfo
2 ORDER BY contact_id ASC

```

The results are displayed in a Data Output grid:

	contact_id [PK] integer	phone character varying (20)	email character varying (100)
1	1	123-456-7890	john.doe@example.com
2	2	987-654-3210	jane.smith@example.com
3	3	111-222-3333	alice.johnson@example.com
4	4	444-555-6666	robert.williams@example.com
5	5	777-888-9999	sophia.garcia@example.com
6	6	111-222-3334	william.brown@example.com
7	7	444-555-6667	olivia.rodriguez@example.com
8	8	777-888-9998	liam.martinez@example.com
9	9	111-222-3335	emma.lee@example.com
10	10	444-555-6668	Noah.Neuman@example.com

Total rows: 100 of 100 Query complete 00:00:00.215 Ln 1, Col 1

-- Home Care Worker Table

CREATE TABLE HomeCareWorker (

Worker_ID SERIAL PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

Qualification VARCHAR(100),

Dept_ID INTEGER,

CONSTRAINT worker_qualification_check CHECK (Qualification <> '')

);

pgAdmin 4

File Object Tools Help

Object Explorer

- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
 - Trigger Functions

homecareworker

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
worker_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('ho')
name	character varying	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
qualification	character varying	100		<input type="checkbox"/>	<input type="checkbox"/>	
dept_id	integer			<input type="checkbox"/>	<input type="checkbox"/>	

X Close Reset Save

Ln 1, Col 1

Type here to search

12:19 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
 - Trigger Functions

homecareworker

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns
homecareworker_pkey	worker_id

X Close Reset Save

Ln 1, Col 1

Type here to search

12:19 17-11-2023

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the 'Tables' section, the 'homecareworker' table is selected. A query window displays the following SQL code:

```

1 SELECT * FROM public.homecareworker
2 ORDER BY worker_id ASC

```

The Data Output tab shows the results of the query:

	worker_id [PK] integer	name character varying (100)	qualification character varying (100)	dept_id integer
1	1	Emily Johnson	Registered Nurse	1
2	2	Michael Brown	Licensed Practical Nurse	2
3	3	Sophia Martinez	Certified Nursing Assistant	1
4	4	William Miller	Registered Nurse	3
5	5	Olivia Garcia	Home Health Aide	2
6	6	James Wilson	Physical Therapist	1
7	7	Ava Rodriguez	Occupational Therapist	3
8	8	Ethan Jones	Speech Language Patholo...	2
9	9	Isabella Taylor	Medical Social Worker	1
10	10	Ian Davis	Registered Nurse	3

Total rows: 100 of 100 Query complete 00:00:00.195

-- Service Table

```

CREATE TABLE Service (
    Service_ID SERIAL PRIMARY KEY,
    ServiceName VARCHAR(100) NOT NULL,
    Duration INTERVAL,
    Cost DECIMAL(10, 2),
    Worker_ID INTEGER,
    Client_ID INTEGER,
    CONSTRAINT service_cost_check CHECK (Cost >= 0)
);

```

pgAdmin 4

File Object Tools Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contacti... pi akrn

service

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
service_id	integer			On	On	nexval('ser')
servicename	character varying	100		On	Off	
duration	interval			Off	Off	
cost	numeric	10	2	Off	Off	
worker_id	integer			Off	Off	
client_id	integer			Off	Off	

i ? Close Reset Save

5 5 2023-11-05 13:00:00 5 5 2
6 6 2023-11-06 14:30:00 6 6 3
7 7 2023-11-07 15:45:00 7 7 1
8 8 2023-11-08 08:30:00 8 8 2
9 9 2023-11-09 09:45:00 9 9 3
10 10 2023-11-10 12:00:00 10 10 1

Total rows: 100 of 100 Query complete 00:00:00.219 Ln 1, Col 1

Type here to search

Windows Taskbar: pgAdmin 4, File Explorer, Google Chrome, FileZilla, Netflix, Python, pgAdmin 4

pgAdmin 4

File Object Tools Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contacti... pi akrn

service

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Columns

Name	Columns
service_pkey	service_id

i ? Close Reset Save

5 5 2023-11-05 13:00:00 5 5 2
6 6 2023-11-06 14:30:00 6 6 3
7 7 2023-11-07 15:45:00 7 7 1
8 8 2023-11-08 08:30:00 8 8 2
9 9 2023-11-09 09:45:00 9 9 3
10 10 2023-11-10 12:00:00 10 10 1

Total rows: 100 of 100 Query complete 00:00:00.219 Ln 1, Col 1

Type here to search

Windows Taskbar: pgAdmin 4, File Explorer, Google Chrome, FileZilla, Netflix, Python, pgAdmin 4

pgAdmin 4

File Object Tools Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contacti... pi akrn

service

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns	Referenced Table
service_client_id_fkey	(client_id) -> (client_id)	public.client
service_worker_id_fkey	(worker_id) -> (worker_id)	public.homecareworker

i **?** **X Close** **Reset** **Save**

5 5 2023-11-05 13:00:00 5 5 2
6 6 2023-11-06 14:30:00 6 6 3
7 7 2023-11-07 15:45:00 7 7 1
8 8 2023-11-08 08:30:00 8 8 2
9 9 2023-11-09 09:45:00 9 9 3
10 10 2023-11-10 12:00:00 10 10 1

Total rows: 100 of 100 Query complete 00:00:00.219 Ln 1, Col 1

Type here to search

Windows Taskbar: Type here to search, File Explorer, Google Chrome, FileZilla, Netflix, Python, pgAdmin 4

12:23 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contacti... pi akrn

public.service

Query History

```
1 SELECT * FROM public.service
2 ORDER BY service_id ASC
```

Data Output Messages Notifications

service_id [PK] Integer	servicename character varying(100)	duration interval	cost numeric(10,2)	worker_id integer	client_id integer
1	Medical Checkup	02:00:00	150.00	1	1
2	Assistance with Medication	01:00:00	80.00	2	2
3	Physical Therapy Session	01:30:00	120.00	6	3
4	Occupational Therapy Session	01:00:00	100.00	7	4
5	Home Health Aide Assistance	02:00:00	90.00	5	5
6	Speech Therapy Session	01:30:00	130.00	8	6
7	Medical Checkup	02:00:00	150.00	9	7
8	Assistance with Medication	01:00:00	80.00	10	8
9	Physical Therapy Session	01:30:00	120.00	11	9
10	Occupational Therapy Session	01:00:00	100.00	12	10

Total rows: 100 of 100 Query complete 00:00:00.346 Ln 1, Col 1

Type here to search

Windows Taskbar: Type here to search, File Explorer, Google Chrome, FileZilla, Netflix, Python, pgAdmin 4

12:23 17-11-2023

-- Schedule Table

CREATE TABLE Schedule (

Schedule_ID SERIAL PRIMARY KEY,

Date DATE,

Time TIME,

Client_ID INTEGER,

Worker_ID INTEGER,

Dept_ID INTEGER

);

The screenshot shows the pgAdmin 4 interface with the 'schedule' table selected in the Object Explorer. The 'Columns' tab is active, displaying the following columns:

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
schedule_id	integer	1		On	On	nextval('sc...
date	date	1		Off	Off	
time	time without time zone	1	0	Off	Off	
client_id	integer	1		Off	Off	
worker_id	integer	1		Off	Off	
dept_id	integer	1		Off	Off	

Below the table definition, there is a preview of 100 rows of data:

Index	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
5	5	2023-01-15	Client responded positively to therapeutic exercises.	5	3	
6	6	2023-01-18	Noted client's remarkable progress in speech and language.	6	4	
7	7	2023-01-20	Observed improvements in motor skills and coordination.	7	1	
8	8	2023-01-22	Client's cognitive function showed noticeable enhancement.	8	2	
9	9	2023-01-25	Reported increased independence in daily activities.	9	6	
10	10	2023-01-28	Noted positive response to medication and care regimen.	10	3	

Total rows: 100 of 100 Query complete 00:00:00.269

The screenshot shows the pgAdmin 4 interface with the 'schedule' table selected in the Object Explorer. The 'Constraints' tab is active, showing a primary key constraint named 'schedule_pkey' defined on the 'schedule_id' column.

Name	Columns
schedule_pkey	schedule_id

Below the table definition, there is a preview of 100 rows of data:

Index	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
5	5	2023-01-15	Client responded positively to therapeutic exercises.	5	3	
6	6	2023-01-18	Noted client's remarkable progress in speech and language.	6	4	
7	7	2023-01-20	Observed improvements in motor skills and coordination.	7	1	
8	8	2023-01-22	Client's cognitive function showed noticeable enhancement.	8	2	
9	9	2023-01-25	Reported increased independence in daily activities.	9	6	
10	10	2023-01-28	Noted positive response to medication and care regimen.	10	3	

Total rows: 100 of 100 Query complete 00:00:00.269

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - < schedule
 - schedule
 - service
 - workhistory
 - Trigger Functions

schedule

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns	Referenced Table
schedule_client_id_fkey	(client_id) -> (client_id)	public.client
schedule_dept_id_fkey	(dept_id) -> (dept_id)	public.department
schedule_worker_id_fkey	(worker_id) -> (worker_id)	public.homecareworker

i **?** **X Close** **Reset** **Save**

5 5 2023-01-15 Client responded positively to therapeutic exercises. 5 3
 6 6 2023-01-18 Noted client's remarkable progress in speech and language. 6 4
 7 7 2023-01-20 Observed improvements in motor skills and coordination. 7 1
 8 8 2023-01-22 Client's cognitive function showed noticeable enhancement. 8 2
 9 9 2023-01-25 Reported increased independence in daily activities. 9 6
 10 10 2023-01-28 Noted positive response to medication and care regimen. 10 3

Total rows: 100 of 100 Query complete 00:00:00.269

Ln 1, Col 1

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - < schedule
 - schedule
 - service
 - workhistory
 - Trigger Functions

public.schedule

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contacti... pi akrn

Query History

```
1 SELECT * FROM public.schedule
2 ORDER BY schedule_id ASC
```

Data Output Messages Notifications

schedule_id	date	time	client_id	worker_id	dept_id
1	2023-11-01	08:00:00	1	1	1
2	2023-11-02	09:30:00	2	2	2
3	2023-11-03	10:45:00	3	3	3
4	2023-11-04	11:15:00	4	4	1
5	2023-11-05	13:00:00	5	5	2
6	2023-11-06	14:30:00	6	6	3
7	2023-11-07	15:45:00	7	7	1
8	2023-11-08	08:30:00	8	8	2
9	2023-11-09	09:45:00	9	9	3
10	2023-11-10	12:00:00	10	10	1

Total rows: 100 of 100 Query complete 00:00:00.219

Ln 1, Col 1

-- Payment Table

CREATE TABLE Payment (

Payment_ID SERIAL PRIMARY KEY,

Amount DECIMAL(10, 2),

Date DATE,

Method VARCHAR(50),

Client_ID INTEGER,

```

Worker_ID INTEGER,
CONSTRAINT payment_amount_check CHECK (Amount >= 0)
);

```

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - < public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - > Tables (11)
 - > client
 - > contactinfo
 - > department
 - > feedback
 - > homecareworker
 - > payment
 - > qualification
 - > report
 - > schedule
 - > service
 - > workhistory
 - > Trigger Functions

payment

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
payment_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('pa...
amount	numeric	10	2	<input type="radio"/>	<input checked="" type="radio"/>	
date	date			<input type="radio"/>	<input checked="" type="radio"/>	
method	character varying	50		<input type="radio"/>	<input checked="" type="radio"/>	
client_id	integer			<input type="radio"/>	<input checked="" type="radio"/>	
worker_id	integer			<input type="radio"/>	<input checked="" type="radio"/>	

i ? Close Reset Save

5 5 Olivia Garcia Home Health Aide 2
6 6 James Wilson Physical Therapist 1
7 7 Ava Rodriguez Occupational Therapist 3
8 8 Ethan Jones Speech-Language Patholo... 2
9 9 Isabella Taylor Medical Social Worker 1
10 10 Liam Davis Registered Nurse 3

Total rows: 100 of 100 Query complete 00:00:00.195

Ln 1, Col 1 12:19 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - < public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - > Tables (11)
 - > client
 - > contactinfo
 - > department
 - > feedback
 - > homecareworker
 - > payment
 - > qualification
 - > report
 - > schedule
 - > service
 - > workhistory
 - > Trigger Functions

payment

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns
payment_pkey	payment_id

i ? Close Reset Save

5 5 Olivia Garcia Home Health Aide 2
6 6 James Wilson Physical Therapist 1
7 7 Ava Rodriguez Occupational Therapist 3
8 8 Ethan Jones Speech-Language Patholo... 2
9 9 Isabella Taylor Medical Social Worker 1
10 10 Liam Davis Registered Nurse 3

Total rows: 100 of 100 Query complete 00:00:00.195

Ln 1, Col 1 12:19 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contacti... pi > atm

payment

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns	Referenced Table
payment_client_id_fkey	(client_id) -> (client_id)	public.client
payment_worker_id_fkey	(worker_id) -> (worker_id)	public.homecareworker

Table Data

5	5	Olivia Garcia	Home Health Aide	2
6	6	James Wilson	Physical Therapist	1
7	7	Ava Rodriguez	Occupational Therapist	3
8	8	Ethan Jones	Speech Language Patholo...	2
9	9	Isabella Taylor	Medical Social Worker	1
10	10	Liam Davis	Registered Nurse	3

Total rows: 100 of 100 Query complete 00:00:00.195 Ln 1, Col 1

12:20 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contacti... pi > atm

Query

```
1 SELECT * FROM public.payment
2 ORDER BY payment_id ASC
```

Data Output

payment_id [PK] integer	amount numeric (10,2)	date date	method character varying (50)	client_id integer	worker_id integer
1	1	120.00	2023-11-01	Credit Card	1
2	2	90.00	2023-11-02	Cash	2
3	3	100.00	2023-11-03	Debit Card	3
4	4	150.00	2023-11-04	Cash	4
5	5	80.00	2023-11-05	Credit Card	5
6	6	130.00	2023-11-06	Debit Card	6
7	7	100.00	2023-11-07	Cash	7
8	8	90.00	2023-11-08	UPI	8
9	9	120.00	2023-11-09	Debit Card	9
10	10	110.00	2023-11-10	Cash	10

Total rows: 100 of 100 Query complete 00:00:00.227 Ln 1, Col 1

12:20 17-11-2023

-- Qualification Table

CREATE TABLE Qualification (

Institution VARCHAR(100),

Degree VARCHAR(100),

Title_Of_Certification VARCHAR(100),

Worker_ID INTEGER PRIMARY KEY

);

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- > Trigger Functions

qualification

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
institution	character varying	100		<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	
degree	character varying	100		<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	
title_of_certification	character varying	100		<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	
worker_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	

5 5 80.00 2023-11-05 Credit Card 5 5
 6 6 130.00 2023-11-06 Debit Card 6 6
 7 7 100.00 2023-11-07 Cash 7 7
 8 8 90.00 2023-11-08 UPI 8 8
 9 9 120.00 2023-11-09 Debit Card 9 9
 10 10 110.00 2023-11-10 Cash 10 10

Total rows: 100 of 100 Query complete 00:00:00.227

Ln 1, Col 1

12:20 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- > Trigger Functions

qualification

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Columns

Name	Columns
qualification_pkey	worker_id

5 5 80.00 2023-11-05 Credit Card 5 5
 6 6 130.00 2023-11-06 Debit Card 6 6
 7 7 100.00 2023-11-07 Cash 7 7
 8 8 90.00 2023-11-08 UPI 8 8
 9 9 120.00 2023-11-09 Debit Card 9 9
 10 10 110.00 2023-11-10 Cash 10 10

Total rows: 100 of 100 Query complete 00:00:00.227

Ln 1, Col 1

12:20 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- < Trigger Functions

qualification

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns	Referenced Table
qualification_worker_id_key	(worker_id) -> (worker_id)	public.homecareworker

i **?** **X Close** **Reset** **Save**

5 5 80.00 2023-11-05 Credit Card 5 5
 6 6 130.00 2023-11-06 Debit Card 6 6
 7 7 100.00 2023-11-07 Cash 7 7
 8 8 90.00 2023-11-08 UPI 8 8
 9 9 120.00 2023-11-09 Debit Card 9 9
 10 10 110.00 2023-11-10 Cash 10 10

Total rows: 100 of 100 Query complete 00:00:00.227

Ln 1, Col 1

Type here to search

Windows Taskbar: pgAdmin 4, File, Object, Tools, Help, Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, Homecare Worker, public.client/H..., public.contacti..., public.payment@PostgreSQL 15, pgAdmin 4, 12:20, 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- < Trigger Functions

qualification

Query History

```
1 SELECT * FROM public.qualification
2 ORDER BY worker_id ASC
```

Data Output Messages Notifications

institution	degree	title_of_certification	worker_id
University of California	Master's Degree	Physical Therapy Certification	1
Stanford University	Bachelor's Degree	Occupational Therapy Certification	2
University of Michigan	Master's Degree	Nursing Certification	3
Harvard University	Doctorate	Medical Doctor Certification	4
Massachusetts Institute of Technology	Bachelor's Degree	Speech-Language Pathology Certification	5
Yale University	Master's Degree	Home Health Aide Certification	6
Columbia University	Doctorate	Physical Therapy Doctorate	7
University of Oxford	Bachelor's Degree	Occupational Therapy Certification	8
University of Cambridge	Master's Degree	Nursing Practitioner Certification	9
Imperial College London	Doctorate	Medical Assistant Doctorate	10

Total rows: 100 of 100 Query complete 00:00:00.207

Ln 1, Col 1

Type here to search

Windows Taskbar: pgAdmin 4, File, Object, Tools, Help, Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, Homecare Worker, public.client/H..., public.contacti..., public.payment@PostgreSQL 15, pgAdmin 4, 12:21, 17-11-2023

-- Work History Table

CREATE TABLE WorkHistory (

Employment_Dates DATE,

JobTitle VARCHAR(100),

Description TEXT,

WorkerID INTEGER PRIMARY KEY

);

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- < Trigger Functions

workhistory

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s) Select to inherit from...

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
employment_dates	character varying	100		<input checked="" type="radio"/>	<input checked="" type="radio"/>	
jobtitle	character varying	100		<input checked="" type="radio"/>	<input checked="" type="radio"/>	
description	text			<input checked="" type="radio"/>	<input checked="" type="radio"/>	
workerid	integer			<input checked="" type="radio"/>	<input checked="" type="radio"/>	

Constraints

Name	Columns
workhistory_pkey	workerid

Total rows: 100 of 100 Query complete 00:00:00.346

Ln 1, Col 1

12:23 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- < Trigger Functions

workhistory

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns
workhistory_pkey	workerid

Total rows: 100 of 100 Query complete 00:00:00.346

Ln 1, Col 1

12:24 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- < Trigger Functions

workhistory

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns	Referenced Table
workhistory_workerid_fkey	(workerid) -> (worker_id)	public.homecareworker

i **?** **X Close** **Reset** **Save**

5	5	Home Health Aide Assistance	02:00:00	90.00	5	5
6	6	Speech Therapy Session	01:30:00	130.00	8	6
7	7	Medical Checkup	02:00:00	150.00	9	7
8	8	Assistance with Medication	01:00:00	80.00	10	8
9	9	Physical Therapy Session	01:00:00	120.00	11	9
10	10	Occupational Therapy Session	01:00:00	100.00	12	10

Total rows: 100 of 100 Query complete 00:00:00.346

Ln 1, Col 1

12:24 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- < Trigger Functions

public.workhistory

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contacti... pi akrn

Query History

```
1 SELECT * FROM public.workhistory
2 ORDER BY workerid ASC
```

Data Output Messages Notifications

	employment_dates	jobtitle	description	workerid
1	character varying (100)	character varying (100)	text	[PK] integer
2	2018-01-01 to 2020-12-31	Medical Assistant	Assisted doctors and nurses in patient care.	1
3	2019-02-15 to 2021-03-20	Registered Nurse	Provided direct patient care and administered medications.	2
4	2017-09-10 to 2022-01-05	Physical Therapist	Designed and implemented rehabilitation programs.	3
5	2020-05-01 to 2023-06-30	Occupational Therapist	Helped patients with daily living activities and rehabilitation exercises.	4
6	2016-08-12 to 2019-11-30	Home Health Aide	Assisted elderly clients with personal care and household tasks.	5
7	2018-03-20 to 2022-04-25	Speech-Language Pathologist	Evaluated and treated communication and swallowing disorders.	6
8	2019-07-01 to 2023-08-15	Medical Assistant	Performed administrative and clinical tasks in a medical office.	7
9	2017-11-05 to 2020-12-10	Registered Nurse	Managed and coordinated patient care plans.	8
10	2016-10-15 to 2019-12-20	Physical Therapist	Provided physical therapy services and rehabilitation.	9
11	2018-12-01 to 2022-01-31	Occupational Therapist	Assessed and treated patients to enhance their daily living skills.	10

Total rows: 100 of 100 Query complete 00:00:00.642

Ln 1, Col 1

12:24 17-11-2023

-- Department Table

CREATE TABLE Department (

Dept_ID SERIAL PRIMARY KEY,

Dept_Name VARCHAR(100) NOT NULL,

Budget DECIMAL(15, 2),

Contact_Info TEXT

);

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- > Trigger Functions

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contactinfo/H...

department

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
dept_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextrval('de')
dept_name	character varying	100		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
budget	numeric	15	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
contact_info	text			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Table Data

dept_id	dept_name	budget	contact_info
5	5	777-888-9999	sophiagarcia@example.com
6	6	111-222-3334	williambrown@example.com
7	7	444-555-6667	oliviarodriguez@example.com
8	8	777-888-9998	liammartinez@example.com
9	9	111-222-3335	emmalee@example.com
10	10	444-555-6668	noahneuwen@example.com

Total rows: 100 of 100 Query complete 00:00:00.215

X Close Reset Save

Ln 1, Col 1

Type here to search

Windows Taskbar: pgAdmin 4, File Explorer, Google Chrome, Mozilla Firefox, Netflix, Microsoft Edge, GitHub

System tray: ENG 12:15 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- > Trigger Functions

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contactinfo/H...

department

General Columns Advanced Constraints Parameters Security SQL

Constraints

Name	Columns
department_pkey	dept_id

Table Data

dept_id	dept_name	budget	contact_info
5	5	777-888-9999	sophiagarcia@example.com
6	6	111-222-3334	williambrown@example.com
7	7	444-555-6667	oliviarodriguez@example.com
8	8	777-888-9998	liammartinez@example.com
9	9	111-222-3335	emmalee@example.com
10	10	444-555-6668	noahneuwen@example.com

Total rows: 100 of 100 Query complete 00:00:00.215

X Close Reset Save

Ln 1, Col 1

Type here to search

Windows Taskbar: pgAdmin 4, File Explorer, Google Chrome, Mozilla Firefox, Netflix, Microsoft Edge, GitHub

System tray: ENG 12:17 17-11-2023

The screenshot shows the pgAdmin 4 interface. The left sidebar is titled 'Object Explorer' and lists various database objects under 'Schemas (1) public'. Under 'Tables (11)', the 'department' table is selected. The main pane contains a SQL query window with the following code:

```

1 SELECT * FROM public.department
2 ORDER BY dept_id ASC

```

The results are displayed in a Data Output grid:

	dept_id [PK] integer	dept_name character varying (100)	budget numeric (15,2)	contact_info text
1	1	Physical Therapy Department	50000.00	Contact: (123) 456-7890, Email: ptdepartment@example.com
2	2	Occupational Therapy Department	60000.00	Contact: (234) 567-8901, Email: otdepartment@example.com
3	3	Nursing Department	70000.00	Contact: (345) 678-9012, Email: nursingdepartment@example.com
4	4	Speech Therapy Department	55000.00	Contact: (456) 789-0123, Email: speechdepartment@example.com
5	5	Medical Assistance Department	48000.00	Contact: (567) 890-1234, Email: madepartment@example.com
6	6	Home Health Aide Department	52000.00	Contact: (678) 901-2345, Email: hhadepartment@example.com
7	7	Administration Department	75000.00	Contact: (789) 012-3456, Email: admindepartment@example.com
8	8	Human Resources Department	65000.00	Contact: (890) 123-4567, Email: hrdepartment@example.com
9	9	Training and Development Department	60000.00	Contact: (901) 234-5678, Email: trainingdepartment@example.com
10	10	Quality Assurance Department	58000.00	Contact: (012) 345-6789, Email: qadepartment@example.com

Total rows: 100 of 100 Query complete 00:00:00.237

-- Report Table

CREATE TABLE Report (

Report_ID SERIAL PRIMARY KEY,

ReportDate DATE,

Description TEXT,

Client_ID INTEGER,

Dept_ID INTEGER

);

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- > Trigger Functions

report

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
report_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextrval('req')
reportdate	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
description	text			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
client_id	integer			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
dept_id	integer			<input type="checkbox"/>	<input checked="" type="checkbox"/>	

i ? Close Reset Save

5 Massachusetts Institute of Technology Bachelor's Degree Speech-Language Pathology Certification
 6 Yale University Master's Degree Home Health Aide Certification
 7 Columbia University Doctorate Physical Therapy Doctorate
 8 University of Oxford Bachelor's Degree Occupational Therapy Certification
 9 University of Cambridge Master's Degree Nursing Practitioner Certification
 10 Imperial College London Doctorate Medical Assistant Doctorate

Total rows: 100 of 100 Query complete 00:00:207 Ln 1, Col 1

12:21 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- > Trigger Functions

report

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name Columns

report_pkey	report_id
-------------	-----------

i ? Close Reset Save

5 Massachusetts Institute of Technology Bachelor's Degree Speech-Language Pathology Certification
 6 Yale University Master's Degree Home Health Aide Certification
 7 Columbia University Doctorate Physical Therapy Doctorate
 8 University of Oxford Bachelor's Degree Occupational Therapy Certification
 9 University of Cambridge Master's Degree Nursing Practitioner Certification
 10 Imperial College London Doctorate Medical Assistant Doctorate

Total rows: 100 of 100 Query complete 00:00:207 Ln 1, Col 1

12:21 17-11-2023

pgAdmin 4

File Object Tools Help

Object Explorer

report

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns	Referenced Table
report_client_id_fkey	(client_id) -> (client_id)	public.client
report_dept_id_fkey	(dept_id) -> (dept_id)	public.department

5 Massachusetts Institute of Technology Bachelor's Degree Speech-Language Pathology Certification
 6 Yale University Master's Degree Home Health Aide Certification
 7 Columbia University Doctorate Physical Therapy Doctorate
 8 University of Oxford Bachelor's Degree Occupational Therapy Certification
 9 University of Cambridge Master's Degree Nursing Practitioner Certification
 10 Imperial College London Doctorate Medical Assistant Doctorate

Total rows: 100 of 100 Query complete 00:00:00.207

Ln 1, Col 1

pgAdmin 4

File Object Tools Help

Object Explorer

report

Query Query History

```
1 SELECT * FROM public.report
2 ORDER BY report_id ASC
```

Data Output Messages Notifications

report_id	reportdate	description	client_id	dept_id
1	2023-01-05	Client showed significant improvement in motor skills.	1	1
2	2023-01-08	Noted progress in speech development during therapy ses...	2	4
3	2023-01-10	Client exhibited enhanced cognitive abilities after treatment...	3	2
4	2023-01-12	Reported improvement in daily living tasks and mobility.	4	6
5	2023-01-15	Client responded positively to therapeutic exercises.	5	3
6	2023-01-18	Noted client's remarkable progress in speech and language.	6	4
7	2023-01-20	Observed improvements in motor skills and coordination.	7	1
8	2023-01-22	Client's cognitive function showed noticeable enhancement.	8	2
9	2023-01-25	Reported increased independence in daily activities.	9	6
10	2023-01-28	Noted positive response to medication and care regimen.	10	3

Total rows: 100 of 100 Query complete 00:00:00.269

Ln 1, Col 1

-- Feedback Table

CREATE TABLE Feedback (

FB_ID SERIAL PRIMARY KEY,

Comment TEXT,

Rating INTEGER,

Worker_ID INTEGER,

Client_ID INTEGER

);

The screenshot shows the pgAdmin 4 interface with the 'feedback' table selected in the Object Explorer. The 'Columns' tab is active, displaying five columns: fb_id, comment, rating, worker_id, and client_id. The 'fb_id' column is defined as an integer with a default value of nextval('fe...'). The 'comment' column is defined as text. The 'rating' and 'worker_id' columns are defined as integers. The 'client_id' column is also defined as an integer. The 'Advanced' tab is visible at the top of the dialog.

The screenshot shows the pgAdmin 4 interface with the 'feedback' table selected in the Object Explorer. The 'Constraints' tab is active, showing a primary key constraint named 'feedback_pkey' defined on the 'fb_id' column. The 'Primary Key' tab is selected. The 'Advanced' tab is visible at the top of the dialog.

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- > Trigger Functions

feedback

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns	Referenced Table
<input type="text"/> feedback_client_id_fkey	(client_id) -> (client_id)	public.client
<input type="text"/> feedback_worker_id_fkey	(worker_id) -> (worker_id)	public.homecareworker

5	5	Medical Assistance Department	48000.00	Contact: (567) 890-1234, Email: madepartment@example.com	t@example.com
6	6	Home Health Aide Department	52000.00	Contact: (678) 901-2345, Email: hhadepartment@example.com	t@example.com
7	7	Administration Department	75000.00	Contact: (789) 012-3456, Email: admindepartment@example.com	tment@example.com
8	8	Human Resources Department	65000.00	Contact: (900) 123-4567, Email: hredepartment@example.com	tment@example.com
9	9	Training and Development Department	60000.00	Contact: (901) 234-5678, Email: trainingdepartment@example.com	tment@example.com
10	10	Quality Assurance Department	58000.00	Contact: (012) 345-6789, Email: qadepartment@example.com	tment@example.com

Total rows: 100 of 100 | Query complete 00:00:237 | Ln 1, Col 1

Type here to search

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- < Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- < Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
- > Trigger Functions

public.feedback

Dashboard Properties SQL Statistics Dependencies Dependents Homecare Worker... public.client/H... public.contacti...

Query History

```
1 SELECT * FROM public.feedback
2 ORDER BY fb_id ASC
```

Data Output Messages Notifications

fb_id [PK] integer	comment text	rating integer	worker_id integer	client_id integer
1	The worker was very attentive and caring.	4	1	1
2	Excellent service provided by the worker.	5	2	2
3	The worker needs to be more punctual.	2	3	3
4	Very knowledgeable and helpful worker.	5	4	4
5	Satisfactory service, but room for improvement.	3	5	5
6	Extremely satisfied with the worker's dedication.	5	6	6
7	Good communication skills, but needs more expertise.	4	7	7
8	The worker was friendly and professional.	5	8	8
9	Could improve in providing detailed explanations.	2	9	9
10	Adequate service but expected better.	3	10	10

Total rows: 100 of 100 | Query complete 00:00:149 | Ln 1, Col 1

Type here to search

SQL Queries

1. Find clients born after 1990.

```
SELECT * FROM Client WHERE EXTRACT(YEAR FROM DateOfBirth) > 1990;
```

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

- Object Explorer:** Shows the schema structure with 11 tables: client, contactinfo, department, feedback, homecareworker, payment, qualification, report, schedule, service, and workhistory.
- Query Editor:** Contains the SQL query: `SELECT * FROM Client WHERE EXTRACT(YEAR FROM DateOfBirth) > 1990;`
- Data Output:** Displays the results of the query, showing 33 rows of client data. The columns are client_id, name, date_of_birth, contact_id, and address.

	client_id	name	date_of_birth	contact_id	address
1	4	Robert Williams	1995-12-20	4	101 Pine Ave
2	8	Liam Martinez	1998-02-27	8	505 Spruce Ave
3	10	Noah Nguyen	2000-08-05	10	707 Cedar Ave
4	12	Ethan King	1993-01-12	12	909 Elm Ave
5	15	Sophie Wilson	1996-06-25	15	1203 Birch St
6	17	Mia Anderson	2002-07-30	17	1405 Oakwood St
7	20	Benjamin Johnson	1991-11-28	20	1708 Elm Ave
8	22	Amelia Brown	1997-01-20	22	1910 Maple Ave
9	26	Evelyn Nguyen	1994-06-07	26	2314 Cedar Ave
10	28	Avery King	2001-08-08	28	2516 Elm Ave
11	31	Ellie Wilson	1999-05-18	31	2819 Birch St
12	33	Riley Anderson	1992-04-16	33	3021 Oakwood St

- Message Bar:** Shows "Query complete 00:00:00.096".
- System Bar:** Shows the date and time as 17-11-2023 12:37.

2. Calculate the average cost of services offered.

```
SELECT AVG(Cost) AS AverageCost FROM Service;
```

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

- Object Explorer:** Shows the schema structure with 11 tables: client, contactinfo, department, feedback, homecareworker, payment, qualification, report, schedule, service, and workhistory.
- Query Editor:** Contains the SQL query: `SELECT AVG(Cost) AS AverageCost FROM Service;`
- Data Output:** Displays the results of the query, showing 1 row with the average cost.

	averagecost
1	111.70000000000000

- Message Bar:** Shows "Query complete 00:00:00.135".
- System Bar:** Shows the date and time as 17-11-2023 12:40.

3. Get the top 5 highest-rated workers.

```
SELECT * FROM Feedback ORDER BY Rating DESC LIMIT 5;
```

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

- Object Explorer:** Shows the database schema with the "Schemas (1)" node expanded, revealing the "public" schema which contains various tables like client, contactinfo, department, feedback, homecareworker, payment, qualification, report, schedule, service, and workhistory.
- Query Editor:** Displays the SQL query: `SELECT * FROM Feedback ORDER BY Rating DESC LIMIT 5;`
- Data Output:** Shows the results of the query as a table:

fb_id	comment	rating	worker_id	client_id
1	Extremely satisfied with the worker's dedication and professionalism.	5	6	6
2	The worker was friendly and professional.	5	8	8
3	Very knowledgeable and helpful worker.	5	4	4
4	Excellent service provided by the worker.	5	2	2
5	Impressed by the worker's commitment and skills.	5	11	11

Total rows: 5 of 5 | Query complete 00:00:00.065 | Ln 2, Col 1

4. Retrieve services costing more than ₹100.

```
SELECT * FROM Service WHERE Cost > 100;
```

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

- Object Explorer:** Shows the database schema with the "Schemas (1)" node expanded, revealing the "public" schema which contains various tables like client, contactinfo, department, feedback, homecareworker, payment, qualification, report, schedule, service, and workhistory.
- Query Editor:** Displays the SQL query: `SELECT * FROM Service WHERE Cost > 100;`
- Data Output:** Shows the results of the query as a table:

service_id	servicename	duration	cost	worker_id	client_id
1	Medical Checkup	02:00:00	150.00	1	1
2	Physical Therapy Session	01:30:00	120.00	6	3
3	Speech Therapy Session	01:30:00	130.00	8	6
4	Medical Checkup	02:00:00	150.00	9	7
5	Physical Therapy Session	01:30:00	120.00	11	9
6	Speech Therapy Session	01:30:00	130.00	14	12
7	Medical Checkup	02:00:00	150.00	15	13
8	Physical Therapy Session	01:30:00	120.00	17	15
9	Speech Therapy Session	01:30:00	130.00	20	18
10	Medical Checkup	02:00:00	150.00	21	19
11	Physical Therapy Session	01:30:00	120.00	23	21
12	Speech Therapy Session	01:30:00	130.00	26	24

Total rows: 50 of 50 | Query complete 00:00:00.107 | Ln 1, Col 40

5. Calculate the total amount paid by clients.

```
SELECT SUM(Amount) AS TotalAmountPaid FROM Payment;
```

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying the schema structure with various objects like Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and 11 Tables. The right pane is the Query Editor, showing the following SQL query:

```
1 SELECT AVG(Cost) AS AverageCost FROM Service;
```

The Data Output tab shows the results of the query:

averagecost	numeric
1	111.70000000000000

Total rows: 1 of 1 | Query complete 00:00:00.135 | Ln 2, Col 1

6. Get the youngest and oldest clients.

```
SELECT * FROM Client
```

```
WHERE DateOfBirth IN (
```

```
SELECT MIN(DateOfBirth) FROM Client
```

```
UNION ALL
```

```
SELECT MAX(DateOfBirth) FROM Client
```

```
);
```

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (1)
 - > public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
- > Tables (11)
 - > client
 - > contactinfo
 - > department
 - > feedback
 - > homecareworker
 - > payment
 - > qualification
 - > report
 - > schedule
 - > service
 - > workhistory
- > Trigger Functions

Query Query History

```

1 SELECT * FROM Client
2 WHERE Date_of_Birth IN (
3   SELECT MIN(Date_of_Birth) FROM Client
4   UNION ALL
5   SELECT MAX(Date_of_Birth) FROM Client
6 );
7
  
```

Data Output Messages Notifications

client_id [PK] integer	name character varying (100)	date_of_birth date	contact_id integer	address text
1	36 Lincoln Johnson	2003-09-09	36	3324 Elm Ave
2	62 Isaac Moore	1966-02-22	62	5950 Maple Ave

Total rows: 2 of 2 Query complete 00:00:00.055 Ln 7, Col 1

Type here to search

7. Find services with a duration longer than 1 hour.

`SELECT * FROM Service WHERE Duration > INTERVAL '1 hour';`

pgAdmin 4

File Object Tools Help

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (1)
 - > public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
- > Tables (11)
 - > client
 - > contactinfo
 - > department
 - > feedback
 - > homecareworker
 - > payment
 - > qualification
 - > report
 - > schedule
 - > service
 - > workhistory
- > Trigger Functions

Query Query History

```

1 SELECT * FROM Service WHERE Duration > INTERVAL '1 hour';
2
  
```

Data Output Messages Notifications

service_id [PK] integer	servicename character varying (100)	duration interval	cost numeric (10,2)	worker_id integer	client_id integer
1	Medical Checkup	02:00:00	150.00	1	1
2	Physical Therapy Session	01:30:00	120.00	6	3
3	Home Health Aide Assistance	02:00:00	90.00	5	5
4	Speech Therapy Session	01:30:00	130.00	8	6
5	Medical Checkup	02:00:00	150.00	9	7
6	Physical Therapy Session	01:30:00	120.00	11	9
7	Home Health Aide Assistance	02:00:00	90.00	13	11
8	Speech Therapy Session	01:30:00	130.00	14	12
9	Medical Checkup	02:00:00	150.00	15	13
10	Physical Therapy Session	01:30:00	120.00	17	15
11	Home Health Aide Assistance	02:00:00	90.00	19	17
12	Speech Therapy Session	01:30:00	130.00	20	18

Total rows: 66 of 66 Query complete 00:00:00.063 Ln 2, Col 1

Type here to search

8. Calculate the average rating given by clients.

`SELECT AVG(Rating) AS AverageRating FROM Feedback;`

The screenshot shows the pgAdmin 4 interface. The left sidebar is titled 'Object Explorer' and lists various database objects under 'Schemas (1)'. The 'Tables (11)' section contains entries like 'client', 'contactinfo', 'department', 'feedback', 'homecareworker', 'payment', 'qualification', 'report', 'schedule', 'service', and 'workhistory'. The main pane displays a query window with the following SQL code:

```
1 SELECT AVG(Rating) AS AverageRating FROM Feedback;
```

The results table shows one row with the value '3.710000000000000'. Below the table, the status bar indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.160'.

9. Retrieve the most expensive service.

`SELECT * FROM Service ORDER BY Cost DESC LIMIT 1;`

The screenshot shows the pgAdmin 4 interface. The left sidebar is titled 'Object Explorer' and lists various database objects under 'Schemas (1)'. The 'Tables (11)' section contains entries like 'client', 'contactinfo', 'department', 'feedback', 'homecareworker', 'payment', 'qualification', 'report', 'schedule', 'service', and 'workhistory'. The main pane displays a query window with the following SQL code:

```
1 SELECT * FROM Service ORDER BY Cost DESC LIMIT 1;
```

The results table shows one row with the following data:

service_id	servicename	duration	cost	worker_id	client_id
1	Medical Checkup	02:00:00	150.00	1	1

Below the table, the status bar indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.103'.

10. Get the total count of clients.

`SELECT COUNT(*) AS TotalClients FROM Client;`

The screenshot shows the pgAdmin 4 interface. The left sidebar is titled 'Object Explorer' and lists various database objects under 'Schemas (1)'. In the main pane, a query window is open with the following SQL code:

```
1 SELECT COUNT(*) AS TotalClients FROM Client;
```

The results table shows one row with the value 100. The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.046'.

11. Calculate the total amount paid by clients.

`SELECT SUM(Amount) AS TotalAmountPaid FROM Payment;`

The screenshot shows the pgAdmin 4 interface. The left sidebar is titled 'Object Explorer' and lists various database objects under 'Schemas (1)'. In the main pane, a query window is open with the following SQL code:

```
1 SELECT SUM(Amount) AS TotalAmountPaid FROM Payment;
```

The results table shows one row with the value 10820.00. The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.047'.

12. Get the count of services provided by each worker.

`SELECT Worker_ID, COUNT(*) AS ServiceCount`

`FROM Service`

GROUP BY Worker_ID;

The screenshot shows the pgAdmin 4 interface with a dark theme. The left sidebar is titled 'Object Explorer' and lists various database objects under the 'public' schema, including tables like 'client', 'contactinfo', 'department', etc. The main pane is titled 'Query' and contains the following SQL code:

```
1 SELECT Worker_ID, COUNT(*) AS NumServices
2 FROM Service
3 GROUP BY Worker_ID;
4
```

The results are displayed in a table titled 'Data Output' with columns 'worker_id' and 'numservices'. The data shows 12 rows of results:

worker_id	numservices
1	55
2	27
3	23
4	56
5	91
6	58
7	8
8	87
9	74
10	54
11	29
12	71

Below the table, a message bar indicates: 'Successfully run. Total query runtime: 60 msec. 98 rows affected.' The status bar at the bottom right shows 'Ln 4, Col 1' and the date '17-11-2023'.

13. Retrieve the maximum duration of services for each worker.

SELECT Worker_ID, MAX(Duration) AS MaxDuration

FROM Service

GROUP BY Worker_ID;

The screenshot shows the pgAdmin 4 interface with a dark theme. The left sidebar is titled 'Object Explorer' and lists various database objects under the 'public' schema, including tables like 'client', 'contactinfo', 'department', etc. The main pane is titled 'Query' and contains the following SQL code:

```
1 SELECT Worker_ID, MAX(Duration) AS MaxDuration
2 FROM Service
3 GROUP BY Worker_ID;
4
5
```

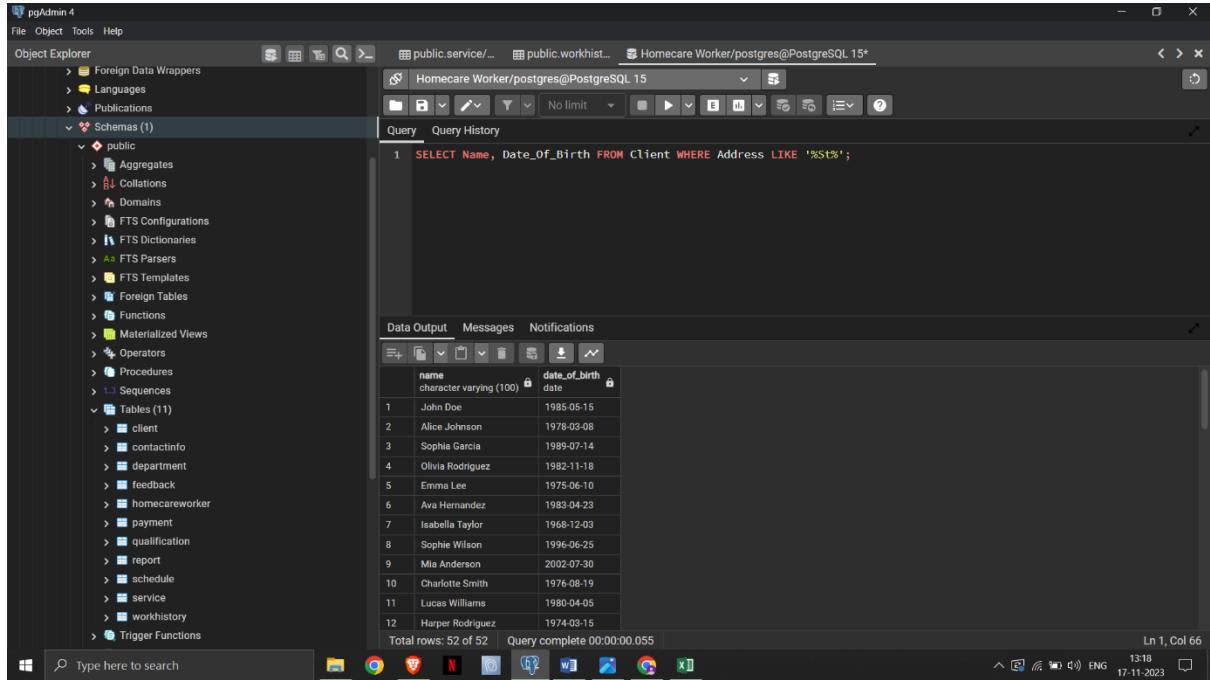
The results are displayed in a table titled 'Data Output' with columns 'worker_id' and 'maxduration'. The data shows 12 rows of results:

worker_id	maxduration
1	02:00:00
2	02:00:00
3	01:30:00
4	01:30:00
5	02:00:00
6	01:00:00
7	01:30:00
8	02:00:00
9	01:30:00
10	01:00:00
11	01:30:00
12	01:30:00

Below the table, a message bar indicates: 'Successfully run. Total query runtime: 60 msec. 98 rows affected.' The status bar at the bottom right shows 'Ln 5, Col 1' and the date '17-11-2023'.

14. Retrieve the client names and dates of birth for clients in "St" from the Client table.

```
SELECT Name, Date_Of_Birth FROM Client WHERE Address LIKE '%St%';
```



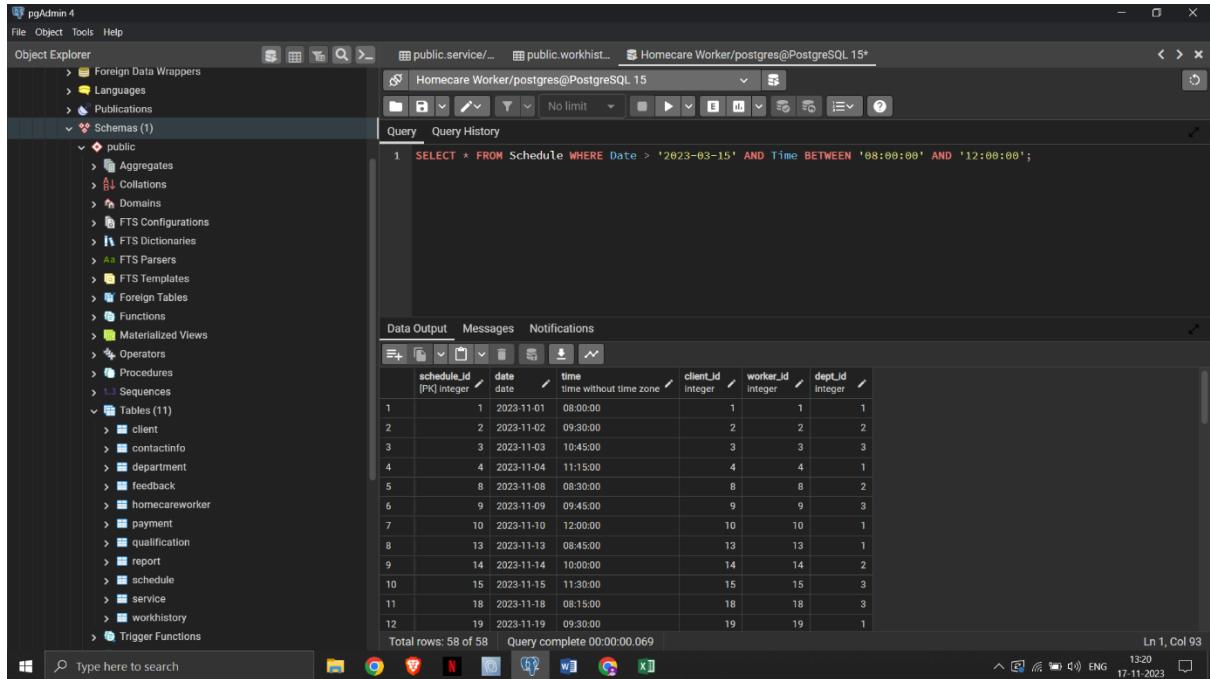
```
1 SELECT Name, Date_of_Birth FROM Client WHERE Address LIKE '%St%';
```

	name	date_of_birth
1	John Doe	1985-05-15
2	Alice Johnson	1978-03-08
3	Sophia Garcia	1989-07-14
4	Olivia Rodriguez	1992-11-16
5	Emma Lee	1975-06-10
6	Ava Hernandez	1983-04-23
7	Isabella Taylor	1968-12-03
8	Sophie Wilson	1996-06-25
9	Mia Anderson	2002-07-30
10	Charlotte Smith	1976-08-19
11	Lucas Williams	1980-04-05
12	Harper Rodriguez	1974-03-15

Total rows: 52 of 52 Query complete 00:00:00.055 Ln 1, Col 66

15. Retrieve the schedules with dates after '2023-03-15' and times between '08:00:00' and '12:00:00' from the Schedule table.

```
SELECT * FROM Schedule WHERE Date > '2023-03-15' AND Time BETWEEN '08:00:00' AND '12:00:00';
```



```
1 SELECT * FROM Schedule WHERE Date > '2023-03-15' AND Time BETWEEN '08:00:00' AND '12:00:00';
```

	schedule_id	date	time	client_id	worker_id	dept_id
1	1	2023-11-01	08:00:00	1	1	1
2	2	2023-11-02	09:30:00	2	2	2
3	3	2023-11-03	10:45:00	3	3	3
4	4	2023-11-04	11:15:00	4	4	1
5	8	2023-11-08	08:30:00	8	8	2
6	9	2023-11-09	09:45:00	9	9	3
7	10	2023-11-10	12:00:00	10	10	1
8	13	2023-11-13	08:45:00	13	13	1
9	14	2023-11-14	10:00:00	14	14	2
10	15	2023-11-15	11:30:00	15	15	3
11	18	2023-11-18	08:15:00	18	18	3
12	19	2023-11-19	09:30:00	19	19	1

Total rows: 58 of 58 Query complete 00:00:00.069 Ln 1, Col 93

16. Retrieve qualifications from institutions containing 'University' and degrees containing 'Bachelor' from the Qualification table.

```
SELECT * FROM Qualification WHERE Institution LIKE '%University%' AND Degree LIKE '%Bachelor%';
```

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer pane displays various database objects like Schemas, Tables, and Functions. The main pane shows a query window with the executed SQL statement and its results. The results are presented in a table format with columns: institution, degree, title_of_certification, and worker_id. The table contains 12 rows of data.

	institution	degree	title_of_certification	worker_id
1	Stanford University	Bachelor's Degree	Occupational Therapy Certification	2
2	University of Oxford	Bachelor's Degree	Occupational Therapy Certification	8
3	University of Pennsylvania	Bachelor's Degree	Physical Therapy Certification	12
4	Tsinghua University	Bachelor's Degree	Physical Therapy Certification	18
5	University of Melbourne	Bachelor's Degree	Medical Assistant Certification	21
6	University of Cape Town	Bachelor's Degree	Occupational Therapy Certification	24
7	University of Cape Town	Bachelor's Degree	Medical Assistant Certification	27
8	University of Pretoria	Bachelor's Degree	Physical Therapy Certification	30
9	University of Pretoria	Bachelor's Degree	Nursing Certification	33
10	Makere University	Bachelor's Degree	Medical Doctor Certification	36
11	University of Dar es Salaam	Bachelor's Degree	Medical Assistant Certification	39
12	University of Cape Town	Bachelor's Degree	Medical Doctor Certification	42

17. Retrieve reports with dates between '2019-03-01' and '2023-08-25' and descriptions containing 'progress' from the Report table.

```
SELECT * FROM Report WHERE ReportDate BETWEEN '2019-03-01' AND '2023-08-25' AND Description LIKE '%progress%';
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
 - Trigger Functions

Query Query History

```
1 SELECT * FROM Report WHERE ReportDate BETWEEN '2019-03-01' AND '2023-08-25' AND Description LIKE '%progress%';
```

Data Output Messages Notifications

reportId	reportdate	description	client_Id	dept_Id
1	2023-01-08	Noted progress in speech development during therapy session...	2	4
2	2023-01-18	Noted client's remarkable progress in speech and language.	6	4
3	2023-02-02	Client demonstrated remarkable progress in motor skills.	11	1
4	2023-03-05	Reported significant progress in motor skills development.	21	1
5	2023-03-28	Noted sustained progress and positive response to care.	30	3
6	2023-04-18	Observed significant progress in speech and communication.	36	4
7	2023-05-05	Reported significant progress in motor skills development.	41	1
8	2023-05-28	Noted sustained progress and positive response to care.	50	3
9	2023-06-18	Observed significant progress in speech and communication.	56	4
10	2023-07-05	Reported significant progress in motor skills development.	61	1
11	2023-07-28	Noted sustained progress and positive response to care.	70	3
12	2023-08-18	Observed significant progress in speech and communication.	76	4

Total rows: 12 of 12 Query complete 00:00:00.244 Ln 1, Col 111

1604 17-11-2023

18. Retrieve phone numbers and unique email addresses of contact info for clients in "Ave" from the Client table.

```
SELECT C.Phone, C.Email FROM ContactInfo C INNER JOIN Client Cl ON C.Contact_ID = Cl.Contact_ID WHERE Cl.Address LIKE '%Ave%';
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- Tables (11)
 - client
 - contactinfo
 - department
 - feedback
 - homecareworker
 - payment
 - qualification
 - report
 - schedule
 - service
 - workhistory
 - Trigger Functions

Query Query History

```
1 C.Phone, C.Email FROM ContactInfo C INNER JOIN Client Cl ON C.Contact_ID = Cl.Contact_ID WHERE Cl.Address LIKE '%Ave%';
```

Data Output Messages Notifications

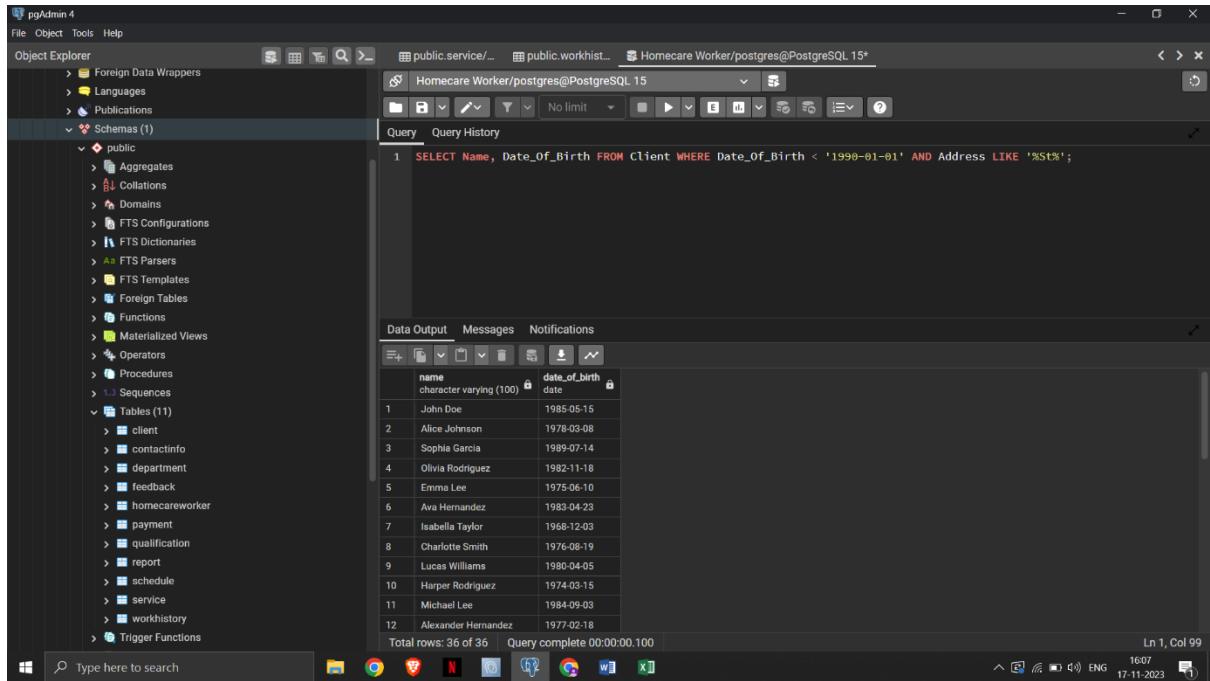
phone	email
987-654-3210	jane.smith@example.com
444-555-6666	robert.williams@example.com
111-222-3334	william.brown@example.com
777-888-9998	liam.martinez@example.com
444-555-6668	noah.nguyen@example.com
987-654-3211	ethan.king@example.com
444-555-6669	mason.more@example.com
111-222-3337	jacob.white@example.com
777-888-9991	james.garcia@example.com
444-555-6671	benjamin.johnson@example.com
987-654-3212	amelia.brown@example.com
444-555-6672	mason.martinez@example.com

Total rows: 48 of 48 Query complete 00:00:00.177 Ln 1, Col 127

1606 17-11-2023

19. Retrieve the client names and dates of birth for clients born before '1990-01-01' and with addresses in "St" from the Client table.

```
SELECT Name, Date_Of_Birth FROM Client WHERE Date_Of_Birth < '1990-01-01' AND Address LIKE '%St%';
```



The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying various database objects like Schemas, Tables, and Functions. The right pane is the Query Editor, where the following SQL query is run:

```
1 SELECT Name, Date_Of_Birth FROM Client WHERE Date_Of_Birth < '1990-01-01' AND Address LIKE '%St%';
```

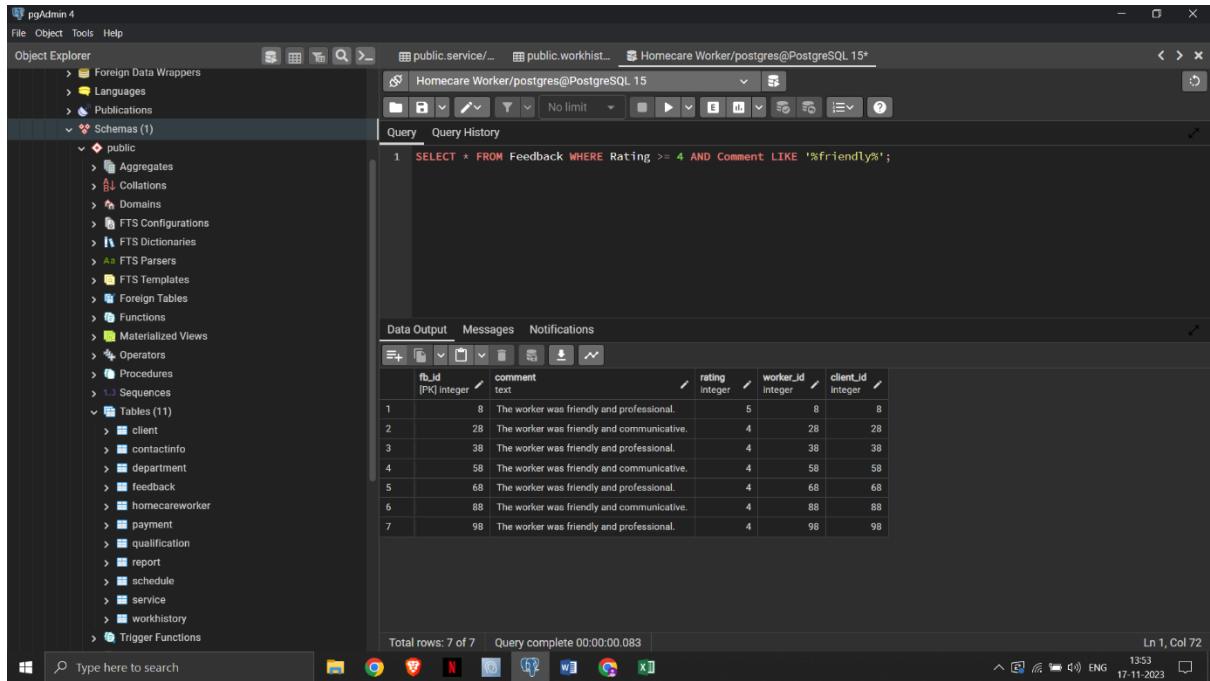
The Data Output tab shows the results:

	name	date_of_birth
1	John Doe	1985-05-15
2	Alice Johnson	1978-03-08
3	Sophia Garcia	1989-07-14
4	Olivia Rodriguez	1982-11-18
5	Emma Lee	1975-06-10
6	Ava Hernandez	1983-04-23
7	Isabella Taylor	1968-12-03
8	Charlotte Smith	1976-08-19
9	Lucas Williams	1980-04-05
10	Harper Rodriguez	1974-03-15
11	Michael Lee	1984-09-03
12	Alexander Hernandez	1977-02-18

Total rows: 36 of 36 Query complete 00:00:00.100

20. Retrieve feedback with ratings greater than or equal to 4 and comments containing 'friendly' from the Feedback table.

```
SELECT * FROM Feedback WHERE Rating >= 4 AND Comment LIKE '%friendly%';
```



The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying various database objects like Schemas, Tables, and Functions. The right pane is the Query Editor, where the following SQL query is run:

```
1 SELECT * FROM Feedback WHERE Rating >= 4 AND Comment LIKE '%friendly%';
```

The Data Output tab shows the results:

	fb_id [PK] integer	comment text	rating integer	worker_id integer	client_id integer
1	8	The worker was friendly and professional.	5	8	8
2	28	The worker was friendly and communicative.	4	28	28
3	38	The worker was friendly and professional.	4	38	38
4	58	The worker was friendly and communicative.	4	58	58
5	68	The worker was friendly and professional.	4	68	68
6	88	The worker was friendly and communicative.	4	88	88
7	98	The worker was friendly and professional.	4	98	98

Total rows: 7 of 7 Query complete 00:00:00.083

21. Display workers along with the departments they are associated with.

```
SELECT w.Name, d.Dept_Name  
FROM HomeCareWorker w  
LEFT JOIN Schedule s ON w.Worker_ID = s.Worker_ID  
LEFT JOIN Department d ON s.Dept_ID = d.Dept_ID;
```

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer pane, which lists various database objects like Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (with public selected), Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (11). The Tables node is expanded, showing client, contactinfo, and department tables. The main central area is the Query Editor, titled 'Homecare Worker/postgres@PostgreSQL 15'. It contains the SQL query from above. Below the query is the Data Output pane, which displays the results of the query as a table:

	name	dept_name
1	Emily Johnson	Physical Therapy Department
2	Michael Brown	Occupational Therapy Department
3	Sophia Martinez	Nursing Department
4	William Miller	Physical Therapy Department
5	Olivia Garcia	Occupational Therapy Department
6	James Wilson	Nursing Department
7	Ava Rodriguez	Physical Therapy Department
8	Ethan Jones	Occupational Therapy Department
9	Isabella Taylor	Nursing Department
10	Liam Davis	Physical Therapy Department
11	Mia Hernandez	Occupational Therapy Department

Total rows: 100 of 100 | Query complete 00:00:00.104 | Ln 5, Col 1

22. Get the total payments made by clients born before 1980.

```
SELECT SUM(p.Amount) AS TotalPayments  
FROM Payment p  
JOIN Client c ON p.Client_ID = c.Client_ID  
WHERE EXTRACT(YEAR FROM c.DateOfBirth) < 1980;
```

pgAdmin 4

File Object Tools Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Homecare Worker/postgres@PostgreSQL 15*

```

SELECT SUM(p.Amount) AS TotalPayments
FROM Payment p
JOIN Client c ON p.Client_ID = c.Client_ID
WHERE EXTRACT(YEAR FROM c.DateOfBirth) < 1980;

```

Data Output Messages Notifications

	totalpayments	numeric
1		4175.00

Total rows: 1 of 1 | Query complete 00:00:00.080 | Ln 5, Col 1

23. List clients who have paid more than the average payment amount.

`SELECT c.Name, p.Amount`

`FROM Client c`

`JOIN Payment p ON c.Client_ID = p.Client_ID`

`WHERE p.Amount > (SELECT AVG(Amount) FROM Payment);`

pgAdmin 4

File Object Tools Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Homecare Worker/postgres@PostgreSQL 15*

```

SELECT c.Name, p.Amount
FROM Client c
JOIN Payment p ON c.Client_ID = p.Client_ID
WHERE p.Amount > (SELECT AVG(Amount) FROM Payment);

```

Data Output Messages Notifications

	name	amount
1	John Doe	120.00
2	Robert Williams	150.00
3	William Brown	130.00
4	Emma Lee	120.00
5	Noah Nguyen	110.00
6	Ethan King	140.00
7	Sophie Wilson	125.00
8	James Garcia	110.00
9	Benjamin Johnson	130.00
10	Lucas Williams	120.00
11	Mason Martinez	115.00

Total rows: 52 of 52 | Query complete 00:00:00.062 | Ln 5, Col 1

24. Display the average rating for each worker.

`SELECT w.Name, AVG(f.Rating) AS AverageRating`

`FROM HomeCareWorker w`

`JOIN Feedback f ON w.Worker_ID = f.Worker_ID`

`GROUP BY w.Name;`

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

- Object Explorer:** Shows the database schema with various objects like Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), Tables (11), and specific tables like client, contactinfo, and department.
- Query Editor:** Contains the following SQL query:

```
1 SELECT w.Name, AVG(f.Rating) AS AverageRating
2 FROM HomeCareWorker w
3 JOIN Feedback f ON w.Worker_ID = f.Worker_ID
4 GROUP BY w.Name;
```
- Data Output:** Displays the results of the query in a table format. The table has three columns: name, averaging, and AverageRating. The data shows 11 rows of workers and their average ratings.
- System Status:** Shows the system date and time as 17-11-2023 13:04.

name	averaging	AverageRating
Ethan Jones	5.000000000000000	5.000000000000000
Olivia Howard	2.000000000000000	2.000000000000000
Avery Powell	4.000000000000000	4.000000000000000
Luna Carter	3.000000000000000	3.000000000000000
Noah Morgan	3.000000000000000	3.000000000000000
Ella Parker	5.000000000000000	5.000000000000000
Nathan Cox	4.000000000000000	4.000000000000000
Noah Perry	4.000000000000000	4.000000000000000
Ellie Hill	4.000000000000000	4.000000000000000
Michael Young	4.000000000000000	4.000000000000000
Evelyn Cook	2.000000000000000	2.000000000000000

25. Find workers who have provided the most expensive service

`SELECT w.Name, s.ServiceName, s.Cost`

`FROM HomeCareWorker w`

`JOIN Service s ON w.Worker_ID = s.Worker_ID`

`WHERE s.Cost = (SELECT MAX(Cost) FROM Service);`

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

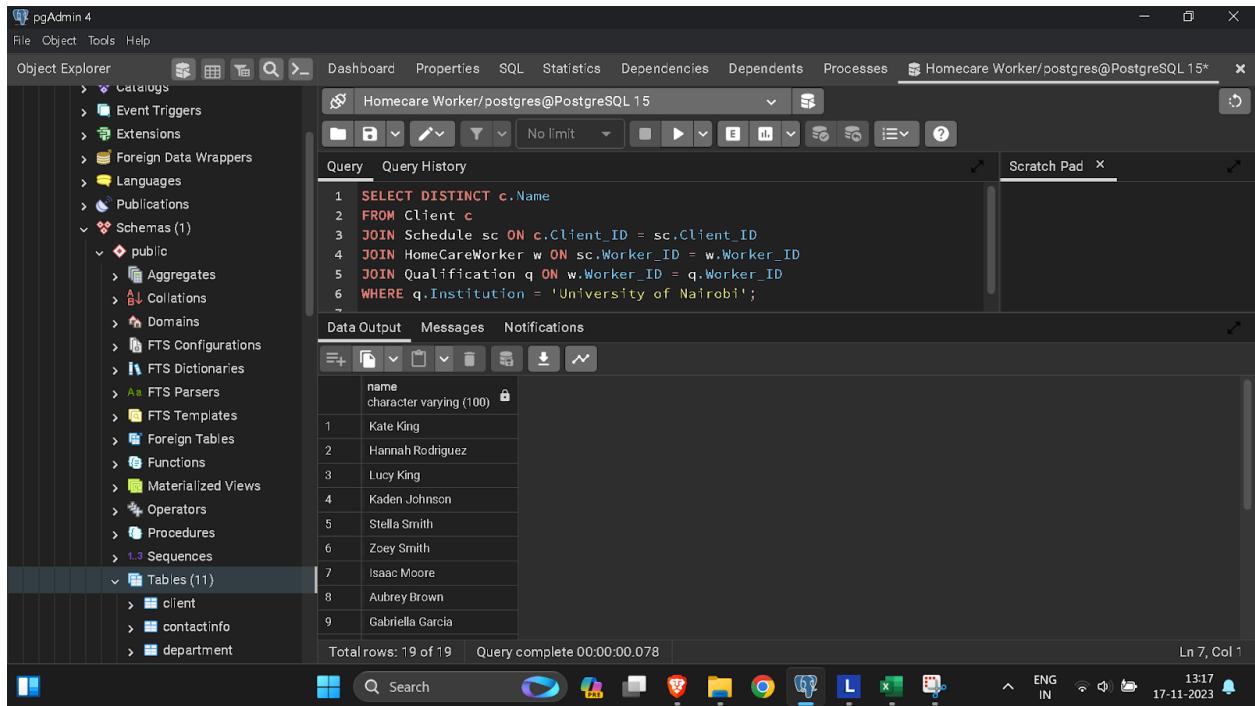
- Object Explorer:** Shows the database schema with various objects like Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), Tables (11), and specific tables like client, contactinfo, and department.
- Query Editor:** Contains the following SQL query:

```
1 SELECT w.Name, s.ServiceName, s.Cost
2 FROM HomeCareWorker w
3 JOIN Service s ON w.Worker_ID = s.Worker_ID
4 WHERE s.Cost = (SELECT MAX(Cost) FROM Service);
```
- Data Output:** Displays the results of the query in a table format. The table has three columns: name, servicename, and cost. The data shows 11 rows of workers and their service costs, all of which are 150.00.
- System Status:** Shows the system date and time as 17-11-2023 13:09.

name	servicename	cost
Emily Johnson	Medical Checkup	150.00
Isabella Taylor	Medical Checkup	150.00
Charlotte Clark	Medical Checkup	150.00
Jack Wright	Medical Checkup	150.00
Ellie Hill	Medical Checkup	150.00
Levi Reed	Medical Checkup	150.00
David Cox	Medical Checkup	150.00
Leo Turner	Medical Checkup	150.00
Carter Hayes	Medical Checkup	150.00
Nathan Cox	Medical Checkup	150.00
Mason Foster	Medical Checkup	150.00

26.List clients who have received services from workers with specific qualifications.

```
SELECT DISTINCT c.Name
FROM Client c
JOIN Schedule sc ON c.Client_ID = sc.Client_ID
JOIN HomeCareWorker w ON sc.Worker_ID = w.Worker_ID
JOIN Qualification q ON w.Worker_ID = q.Worker_ID
WHERE q.Institution = 'University of Nairobi';
```



The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying various database objects like Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), public, Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (11). The right pane contains a query editor window titled 'Homecare Worker/postgres@PostgreSQL 15*' with the following SQL code:

```
1 SELECT DISTINCT c.Name
2 FROM Client c
3 JOIN Schedule sc ON c.Client_ID = sc.Client_ID
4 JOIN HomeCareWorker w ON sc.Worker_ID = w.Worker_ID
5 JOIN Qualification q ON w.Worker_ID = q.Worker_ID
6 WHERE q.Institution = 'University of Nairobi';
```

Below the query editor is a Data Output grid showing the results:

name
Kate King
Hannah Rodriguez
Lucy King
Kaden Johnson
Stella Smith
Zoey Smith
Isaac Moore
Aubrey Brown
Gabriella Garcia

Total rows: 19 of 19 | Query complete 00:00:00.078 | Ln 7, Col 1

27.. Get the top 5 clients who spent the most on services.

```
SELECT c.Name, SUM(p.Amount) AS TotalAmount
FROM Client c
JOIN Payment p ON c.Client_ID = p.Client_ID
GROUP BY c.Name
ORDER BY TotalAmount DESC
LIMIT 5;
```

pgAdmin 4

File Object Tools Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Homecare Worker/postgres@PostgreSQL 15*

Query Scratch Pad

```

SELECT c.Name, SUM(p.Amount) AS TotalAmount
FROM Client c
JOIN Payment p ON c.Client_ID = p.Client_ID
GROUP BY c.Name
ORDER BY TotalAmount DESC
LIMIT 5;

```

Data Output Messages Notifications

	name	totalamount
1	Robert Williams	150.00
2	Stella Smith	140.00
3	Ethan King	140.00
4	Mya Williams	140.00
5	Kate King	140.00

Total rows: 5 of 5 Query complete 00:00:00.078 Ln 7, Col 1

28. Retrieve workers who have provided the least expensive service.

```

SELECT w.Name, s.ServiceName, s.Cost
FROM HomeCareWorker w
JOIN Service s ON w.Worker_ID = s.Worker_ID
WHERE s.Cost = (SELECT MIN(Cost) FROM Service);

```

pgAdmin 4

File Object Tools Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Homecare Worker/postgres@PostgreSQL 15*

Query Scratch Pad

```

SELECT w.Name, s.ServiceName, s.Cost
FROM HomeCareWorker w
JOIN Service s ON w.Worker_ID = s.Worker_ID
WHERE s.Cost = (SELECT MIN(Cost) FROM Service);

```

Data Output Messages Notifications

	name	servicename	cost
1	Michael Brown	Assistance with Medication	80.00
2	Liam Davis	Assistance with Medication	80.00
3	Benjamin Lee	Assistance with Medication	80.00
4	Harper King	Assistance with Medication	80.00
5	Logan Ward	Assistance with Medication	80.00
6	Stella Coleman	Assistance with Medication	80.00
7	Ava Gray	Assistance with Medication	80.00
8	Grace Ross	Assistance with Medication	80.00
9	Scarlett Cooper	Assistance with Medication	80.00

Total rows: 17 of 17 Query complete 00:00:00.065 Ln 4, Col 48

29. List clients who have given feedback with ratings less than 3.

```
SELECT DISTINCT c.Name
```

```

FROM Client c
JOIN Feedback f ON c.Client_ID = f.Client_ID
WHERE f.Rating < 3;

```

The screenshot shows the pgAdmin 4 interface with a query window containing the following SQL code:

```

1 SELECT DISTINCT c.Name
2 FROM Client c
3 JOIN Feedback f ON c.Client_ID = f.Client_ID
4 WHERE f.Rating < 3;

```

The results pane displays the following data:

	name
1	Kayla White
2	Liam Hernandez
3	Gabriel Anderson
4	Zoe Wilson
5	Riley Anderson
6	Emery Johnson
7	Audrey Williams
8	Isabelle Nguyen
9	Leo Rodriguez
10	Harper Rodriguez
11	Mya Williams

Total rows: 20 of 20 | Query complete 00:00:00.067 | Ln 5, Col 1

30. Find clients who have paid more than the average cost of all services.

```

SELECT DISTINCT c.Name
FROM Client c
JOIN Payment p ON c.Client_ID = p.Client_ID
WHERE p.Amount > (SELECT AVG(Cost) FROM Service);

```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with various database objects like Catalogs, Event Triggers, Extensions, Languages, Publications, Schemas (1), and Tables (11). The main pane shows a query results grid for a SELECT statement:

```

1 SELECT DISTINCT c.Name
2 FROM Client c
3 JOIN Payment p ON c.Client_ID = p.Client_ID
4 WHERE p.Amount > (SELECT AVG(Cost) FROM Service);

```

The results show 11 rows of names from the client table:

	name
1	Kate King
2	Ellie Wilson
3	Zoe Wilson
4	Riley Anderson
5	Bailey Lee
6	Aaron Moore
7	Lincoln Johnson
8	Sawyer Brown
9	Lucas Williams
10	William Brown
11	Robert Williams

Total rows: 44 of 44 | Query complete 00:00:00.067 | Ln 5, Col 1

31. Retrieve workers who have received feedback with the highest rating.

```

SELECT w.Name, f.Rating
FROM HomeCareWorker w
JOIN Feedback f ON w.Worker_ID = f.Worker_ID
WHERE f.Rating = (SELECT MAX(Rating) FROM Feedback);

```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with various database objects like Catalogs, Event Triggers, Extensions, Languages, Publications, Schemas (1), and Tables (11). The main pane shows a query results grid for a SELECT statement:

```

1 SELECT w.Name, f.Rating
2 FROM HomeCareWorker w
3 JOIN Feedback f ON w.Worker_ID = f.Worker_ID
4 WHERE f.Rating = (SELECT MAX(Rating) FROM Feedback);

```

The results show 11 rows of names and ratings from the feedback table:

	name	rating
1	Michael Brown	5
2	William Miller	5
3	James Wilson	5
4	Ethan Jones	5
5	Mia Hernandez	5
6	Noah Martin	5
7	Benjamin Lee	5
8	Jack Wright	5
9	Evelyn Green	5
10	Riley Cooper	5
11	Aubrey Brooks	5

Total rows: 31 of 31 | Query complete 00:00:00.048 | Ln 4, Col 53

32. Display services with a duration longer than the average duration of all services.

```
SELECT *
```

FROM Service

WHERE Duration > (SELECT AVG(Duration) FROM Service);

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer tree displays various database objects like Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), public, Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (11). The 'Tables (11)' node is expanded, showing client, contactinfo, and department tables. In the center, a query editor window titled 'Homecare Worker/postgres@PostgreSQL 15' contains the following SQL code:

```
1 SELECT *
2 FROM Service
3 WHERE Duration > (SELECT AVG(Duration) FROM Service);
4
```

Below the query editor is a data grid titled 'Data Output' showing the results of the query. The columns are service_id [PK] integer, servicename character varying (100), duration interval, cost numeric (10,2), worker_id integer, and client_id integer. The data shows 11 rows of service information. At the bottom of the data grid, it says 'Total rows: 66 of 66 Query complete 00:00:00.122 Ln 4, Col 1'. The bottom right corner of the pgAdmin window shows the date and time as '17-11-2023 13:35'.

33. Find the average cost of services provided by workers with the highest qualification.

SELECT AVG(s.Cost) AS AverageCost

FROM Service s

JOIN HomeCareWorker w ON s.Worker_ID = w.Worker_ID

WHERE w.Qualification = (

SELECT MAX(Qualification)

FROM HomeCareWorker

);

pgAdmin 4

File Object Tools Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Homecare Worker/postgres@PostgreSQL 15*

Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1) public Aggregates Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures 1.3 Sequences Tables (11) client contactinfo department

Query Query History

```

1 SELECT AVG(s.Cost) AS AverageCost
2 FROM Service s
3 JOIN HomeCareWorker w ON s.Worker_ID = w.Worker_ID
4 WHERE w.Qualification = (
5     SELECT MAX(Qualification)
6     FROM HomeCareWorker
7 );

```

Data Output Messages Notifications

averagecost	numeric
1	123.333333333333333333

Total rows: 1 of 1 | Query complete 00:00:00.047 | Ln 8, Col 1

13:38 ENG IN 17-11-2023

34. Retrieve workers who have scheduled services for clients of the same age as the worker.

`SELECT w.Name`

`FROM HomeCareWorker w`

`JOIN Schedule sc ON w.Worker_ID = sc.Worker_ID`

`JOIN Client c ON sc.Client_ID = c.Client_ID`

`WHERE EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM w.DateOfBirth) =`

`EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM c.DateOfBirth);`

pgAdmin 4

File Object Tools Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Homecare Worker/postgres@PostgreSQL 15*

Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1) public Aggregates Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures 1.3 Sequences Tables (11) client contactinfo department

Query Query History

```

1 SELECT w.Name
2 FROM HomeCareWorker w
3 JOIN Schedule sc ON w.Worker_ID = sc.Worker_ID
4 JOIN Client c ON sc.Client_ID = c.Client_ID
5 WHERE EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM w.DateOfBirth) =
6

```

Data Output Messages Notifications

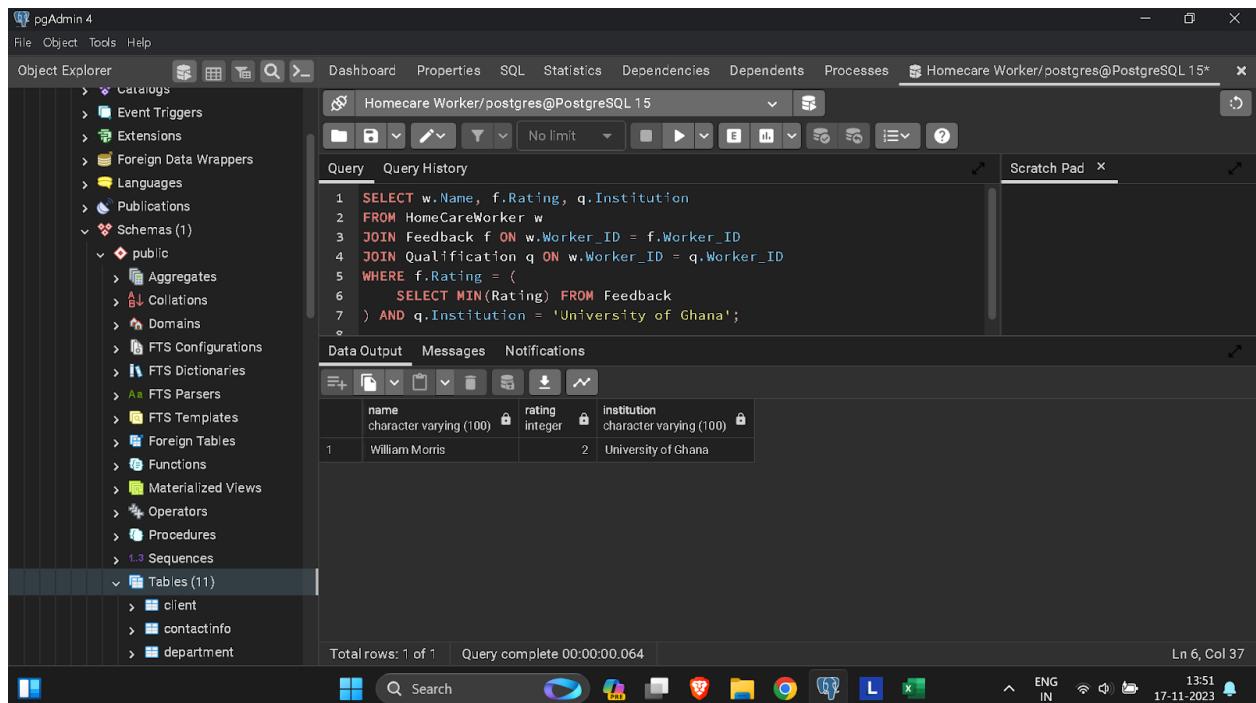
averagecost	numeric
1	123.333333333333333333

Total rows: 1 of 1 | Query complete 00:00:00.047 | Ln 6, Col 1

13:47 ENG IN 17-11-2023

35,Find workers who have received feedback with the lowest rating and have qualifications from University of Ghana.

```
SELECT w.Name, f.Rating, q.Institution
FROM HomeCareWorker w
JOIN Feedback f ON w.Worker_ID = f.Worker_ID
JOIN Qualification q ON w.Worker_ID = q.Worker_ID
WHERE f.Rating = (
    SELECT MIN(Rating) FROM Feedback
) AND q.Institution = 'University of Ghana';
```



The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (with public selected), Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (11). The right pane shows the Query Editor with the SQL query and its execution results. The Data Output tab displays a single row of data:

	name	rating	institution
1	William Morris	2	University of Ghana

Total rows: 1 of 1 | Query complete 00:00:00.064 | Ln 6, Col 37

36.Retrieve clients who have scheduled services from workers with the highest number of qualifications.

```
SELECT c.Name
FROM Client c
JOIN Schedule sc ON c.Client_ID = sc.Client_ID
JOIN HomeCareWorker w ON sc.Worker_ID = w.Worker_ID
JOIN Qualification q ON w.Worker_ID = q.Worker_ID
WHERE w.Worker_ID IN (
    SELECT Worker_ID
    FROM Qualification
    GROUP BY Worker_ID
    ORDER BY COUNT(*) DESC
    LIMIT 1
);
```

```

SELECT c.Name
FROM Client c
JOIN Schedule sc ON c.Client_ID = sc.Client_ID
JOIN HomeCareWorker w ON sc.Worker_ID = w.Worker_ID
JOIN Qualification q ON w.Worker_ID = q.Worker_ID
WHERE w.Worker_ID IN (
    SELECT Worker_ID
    FROM Qualification
    GROUP BY Worker_ID
    ORDER BY COUNT(*) DESC
    LIMIT 1
);

```

Total rows: 1 of 1 | Query complete 00:00:00.075 | Ln 13, Col 1

name
character varying (100)
Eleanor Rodriguez

37. Find workers who have provided services for clients with the lowest and highest ratings in feedback.

```

SELECT w.Name
FROM HomeCareWorker w
JOIN Schedule sc ON w.Worker_ID = sc.Worker_ID
JOIN Client c ON sc.Client_ID = c.Client_ID
JOIN Feedback f ON c.Client_ID = f.Client_ID
WHERE c.Client_ID IN (
    SELECT Client_ID FROM Feedback WHERE Rating =
        SELECT MIN(Rating) FROM Feedback
)
UNION
SELECT Client_ID FROM Feedback WHERE Rating =
    SELECT MAX(Rating) FROM Feedback
);

```

The screenshot shows the pgAdmin 4 interface. The Object Explorer on the left lists various database objects like Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), public, Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (11). The Tables node is expanded, showing client, contactinfo, and department tables. The central area contains a query editor with the following SQL code:

```
1 SELECT w.Name
2 FROM HomeCareWorker w
3 JOIN Schedule sc ON w.Worker_ID = sc.Worker_ID
4 JOIN Client c ON sc.Client_ID = c.Client_ID
5 JOIN Feedback f ON c.Client_ID = f.Client_ID
6 WHERE c.Client_ID IN (
7     SELECT Client_ID FROM Feedback WHERE Rating = (
8         SELECT MIN(Rating) FROM Feedback
9     )
10    UNION
11    SELECT Client_ID FROM Feedback WHERE Rating = (
12        SELECT MAX(Rating) FROM Feedback

```

The Data Output tab shows the results of the query, which are five names: Michael Brown, Sophia Martinez, William Miller, James Wilson, and Ethan Jones.

38. List workers who have provided services for clients in departments with budgets higher than the average department budget.

```
SELECT w.Name
FROM HomeCareWorker w
JOIN Schedule sc ON w.Worker_ID = sc.Worker_ID
JOIN Department d ON sc.Dept_ID = d.Dept_ID
WHERE d.Budget > (
    SELECT AVG(Budget) FROM Department
);
```

```

1 SELECT w.Name
2 FROM HomeCareWorker w
3 JOIN Schedule sc ON w.Worker_ID = sc.Worker_ID
4 JOIN Department d ON sc.Dept_ID = d.Dept_ID
5 WHERE d.Budget > (
6     SELECT AVG(Budget) FROM Department
7 );

```

	name
1	Sophia Martinez
2	James Wilson
3	Isabella Taylor
4	Lucas Perez
5	Charlotte Clark
6	Michael Young
7	Jack Wright
8	Evelyn Green
9	Riley Hill

Total rows: 33 of 33 | Query complete 00:00:00.074 | Ln 8, Col 1

39. Find the names of homecare workers who have qualifications from institutions with the most degrees and have provided services to clients who have received the highest cost service.

```

SELECT H.Name AS HomecareWorkerName
FROM HomeCareWorker H
WHERE H.Worker_ID IN (
    SELECT DISTINCT S.Worker_ID
    FROM Service S
    WHERE S.Cost = (
        SELECT MAX(Cost)
        FROM Service
        WHERE S.Client_ID IN (
            SELECT C.Client_ID
            FROM Client C
            JOIN Service S ON C.Client_ID = S.Client_ID
            WHERE S.Worker_ID IN (
                SELECT DISTINCT Q.Worker_ID
                FROM Qualification Q
                JOIN (
                    SELECT Institution, COUNT(DISTINCT Degree) AS degree_count
                    FROM Qualification
                    GROUP BY Institution
                    ORDER BY degree_count DESC
                    LIMIT 1
                ) AS subquery
                ON Q.Institution = subquery.Institution
            )
        )
    )
)

```

```

) AS subquery ON Q.Institution = subquery.Institution
)
)
)
);

```

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying various database objects like Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), public, Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (11). The right pane contains a query editor window titled 'Homecare Worker/postgres@PostgreSQL 15' with the following SQL code:

```

1 SELECT H.Name AS HomecareWorkerName
2 FROM HomeCareWorker H
3 WHERE H.Worker_ID IN (
4   SELECT DISTINCT S.Worker_ID
5   FROM Service S
6   WHERE S.Cost = (
7     SELECT MAX(Cost)
8     FROM Service
9     WHERE S.Client_ID IN (
10    SELECT C.Client_ID
11    FROM Client C
12    JOIN Service S ON C.Client_ID = S.Client_ID

```

The results pane shows a table with one column 'homecareworkername' containing five rows of data:

homecareworkername
character varying (100)
Ellie Hill
Nathan Cox
Luna Ramirez
Chloe Collins
Emma Henderson

Total rows: 5 of 5 | Query complete 00:00:00.072 | Ln 7, Col 18

40. Retrieve the names of homecare workers who have provided the same service to multiple clients.

```

SELECT H.Name AS HomecareWorkerName
FROM HomeCareWorker H
JOIN Service S ON H.Worker_ID = S.Worker_ID
GROUP BY H.Name
HAVING COUNT(DISTINCT S.Client_ID) > 1;

```

pgAdmin 4

File Object Tools Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Homecare Worker/postgres@PostgreSQL 15*

Query History Scratch Pad

```
1 SELECT H.Name AS HomecareWorkerName
2 FROM HomeCareWorker H
3 JOIN Service S ON H.Worker_ID = S.Worker_ID
4 GROUP BY H.Name
5 HAVING COUNT(DISTINCT S.Client_ID) > 1;
```

Data Output Messages Notifications

homecareworkername
character varying (100)
Emily Johnson
Michael Brown

Total rows: 2 of 2 Query complete 00:00:00.086

Ln 5, Col 41

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying various database objects like Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), public, Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (11). The 'Tables' node is expanded, showing client, contactinfo, and department. The main pane contains a query editor with the following SQL code:

```
1 SELECT H.Name AS HomecareWorkerName
2 FROM HomeCareWorker H
3 JOIN Service S ON H.Worker_ID = S.Worker_ID
4 GROUP BY H.Name
5 HAVING COUNT(DISTINCT S.Client_ID) > 1;
```

Below the query editor is a data output table with one column labeled "homecareworkername". It contains two rows: "Emily Johnson" and "Michael Brown". The status bar at the bottom right indicates "Total rows: 2 of 2" and "Query complete 00:00:00.086". The bottom right corner also shows the date and time: "17-11-2023 14:06".