Documentation

Table: Cities

Purpose of the Table: The Cities table is designed to store information about cities where the company's branches and clients are located. This table provides a structured way to manage city-related data, which can be used for linking branches, clients, and other relevant records in the database, aiding in efficient data retrieval and analysis.

Description of the Table: The Cities table holds information about various cities within Armenia and potentially other provinces. Each row represents a unique city identified by its city_id. The table includes the city's name and the corresponding province, enabling the identification of cities and their locations.

Column Descriptions:

1. **city_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each city. This column automatically increments for each new entry, ensuring that each city has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each city.
2. **city_name (VARCHAR(100)):**
    - **Description:** The name of the city. It is stored as a string with a maximum length of 100 characters.
    - **Purpose:** Provides the name of the city, allowing for easy identification and referencing of city data within the database.
3. **province (VARCHAR(100)):**
    - **Description:** The province or region to which the city belongs. This is also stored as a string with a maximum length of 100 characters.
    - **Purpose:** Helps categorize cities by their respective provinces, enhancing data organization and facilitating regional analysis.

Table: Branches

Purpose of the Table: The Branches table stores detailed information about each branch of the company, including contact details, location, and operational status. This table helps manage branch-specific data, enabling the company to efficiently organize and access information about its various branches.

Description of the Table: The Branches table contains records of all company branches. Each row in the table represents a single branch, identified by a unique branch_id. This table includes the

branch's name, address, city association, contact details, and current status, providing a comprehensive view of the branch's information.

Column Descriptions:

1. **branch_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each branch. This column auto-increments for each new branch entry, ensuring distinct identification.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each branch.
2. **branch_name (VARCHAR(100)):**
    - **Description:** The name of the branch, stored as a string with a maximum length of 100 characters.
    - **Purpose:** Provides the branch's name, facilitating identification and reference within the company's database.
3. **address (VARCHAR(255)):**
    - **Description:** The physical address of the branch, stored as a string with a maximum length of 255 characters.
    - **Purpose:** Specifies the branch's location, aiding in logistical planning and client access.
4. **city_id (INTEGER):**
    - **Description:** A foreign key referencing the city_id from the Cities table, linking each branch to a specific city.
    - **Purpose:** Establishes the relationship between branches and their respective cities, enhancing geographic organization.
5. **phone_number (VARCHAR(20)):**
    - **Description:** The contact phone number of the branch, stored as a string with a maximum length of 20 characters.
    - **Purpose:** Provides direct contact information for the branch, useful for client communication and internal coordination.
6. **email (VARCHAR(100)):**
    - **Description:** The email address of the branch, stored as a string with a maximum length of 100 characters.
    - **Purpose:** Serves as an electronic contact point for the branch, allowing for correspondence and information exchange.
7. **branch_status (VARCHAR(100)):**

- **Description:** The operational status of the branch, such as "Active," "Inactive," or other relevant statuses, stored as a string with a maximum length of 100 characters.
- **Purpose:** Indicates whether the branch is currently operational, helping to manage branch availability and planning.

Table: Clients

Purpose of the Table: The Clients table is designed to store personal and contact information about the company's clients. This table helps manage client-related data, including identification, contact details, and relevant dates, facilitating customer service, marketing, and analytical processes.

Description of the Table: The Clients table contains records of all clients associated with the company. Each row represents a unique client identified by client_id. This table includes client names, contact information, city and branch associations, and gender, along with dates that indicate when the client started or ended their association with the company.

Column Descriptions:

1. **client_id (SERIAL PRIMARY KEY):**
   - **Description:** A unique identifier for each client. This column auto-increments for every new client entry, ensuring each client has a distinct ID.
   - **Purpose:** Serves as the primary key for the table, uniquely identifying each client.
2. **first_name (VARCHAR(50)):**
   - **Description:** The first name of the client, stored as a string with a maximum length of 50 characters.
   - **Purpose:** Captures the client's first name for identification and personalization purposes.
3. **last_name (VARCHAR(50)):**
   - **Description:** The last name of the client, stored as a string with a maximum length of 50 characters.
   - **Purpose:** Captures the client's last name, aiding in full identification.
4. **date_of_birth (DATE):**
   - **Description:** The date of birth of the client, stored in date format.
   - **Purpose:** Used for age verification, personalized marketing, and demographic analysis.
5. **email (VARCHAR(100)):**
   - **Description:** The email address of the client, stored as a string with a maximum length of 100 characters.

- **Purpose:** Serves as the primary electronic contact method for the client, useful for communication and marketing.

6. **phone_number (VARCHAR(20)):**
    - **Description:** The contact phone number of the client, stored as a string with a maximum length of 20 characters.
    - **Purpose:** Provides a direct line of communication with the client.
7. **address (VARCHAR(255)):**
    - **Description:** The physical address of the client, stored as a string with a maximum length of 255 characters.
    - **Purpose:** Specifies the client's residential or contact address, useful for service delivery and correspondence.
8. **city_id (INTEGER):**
    - **Description:** A foreign key referencing the city_id in the Cities table, linking the client to their city of residence.
    - **Purpose:** Establishes the relationship between the client and their city, aiding in location-based analysis.
9. **branch_id (INTEGER):**
    - **Description:** A foreign key referencing the branch_id in the Branches table, associating the client with a specific branch.
    - **Purpose:** Indicates the branch that primarily handles the client, useful for service management.
10. **gender (VARCHAR(10)):**
    - **Description:** The gender of the client, stored as a string with a maximum length of 10 characters (e.g., "Male," "Female").
    - **Purpose:** Used for demographic segmentation and analysis.
11. **start_date (TIMESTAMP):**
    - **Description:** The date and time when the client's information became active or when they first interacted with the company.
    - **Purpose:** Helps track the start of the client relationship, used for historical analysis and service tracking.
12. **end_date (TIMESTAMP):**
    - **Description:** The date and time when the client's information stopped being active or when they ceased interaction with the company.

- **Purpose:** Marks the end of the client relationship, useful for identifying inactive clients and managing records.

Table: Categories

Purpose of the Table: The Categories table is designed to store information about different service categories offered by the company. It organizes services into specific categories, such as skincare, hair removal, or dental services, helping to manage and classify the various services provided.

Description of the Table: The Categories table contains records of all service categories available within the company. Each row represents a unique category identified by category_id. This table includes the category name and the start and end dates that indicate the period during which the category was active.

Column Descriptions:

1. **category_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each service category. This column auto-increments for each new entry, ensuring that each category has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each category.
2. **category_name (VARCHAR(100)):**
    - **Description:** The name of the service category, stored as a string with a maximum length of 100 characters.
    - **Purpose:** Specifies the name of the service category, such as "Skincare," "Hair Removal," or "Dental Services," aiding in the classification and management of services.
3. **start_date (TIMESTAMP):**
    - **Description:** The date and time when the service category became active or was first introduced.
    - **Purpose:** Helps track the introduction of new categories, useful for historical analysis and service timeline management.
4. **end_date (TIMESTAMP):**
    - **Description:** The date and time when the service category was discontinued or stopped being active.
    - **Purpose:** Indicates the end of the category's activity, helping to manage the lifecycle of service categories and identify inactive ones.

Table: Services

Purpose of the Table: The Services table stores detailed information about the various services offered by the company. This table helps manage service-specific data, including pricing, duration, category association, and active status, enabling efficient service management and organization.

Description of the Table: The Services table contains records of all the services provided by the company. Each row represents a unique service identified by service_id. This table includes details such as the service name, price, duration, associated category, and active status, along with timestamps indicating the service's availability period.

Column Descriptions:

1. **service_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each service. This column auto-increments for each new entry, ensuring that each service has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each service.
2. **service_name (VARCHAR(100)):**
    - **Description:** The name of the service, stored as a string with a maximum length of 100 characters.
    - **Purpose:** Provides the service name, aiding in identification and reference within the company's offerings.
3. **price (NUMERIC):**
    - **Description:** The cost of the service, stored as a numeric value.
    - **Purpose:** Specifies the price clients need to pay for the service, essential for billing and pricing analysis.
4. **duration (INTEGER):**
    - **Description:** The duration of the service in minutes, stored as an integer.
    - **Purpose:** Indicates how long the service takes, helping with scheduling and time management.
5. **category_id (INTEGER):**
    - **Description:** A foreign key referencing the category_id in the Categories table, linking each service to its respective category.
    - **Purpose:** Establishes the relationship between services and their categories, aiding in the organization and classification of services.
6. **is_active (BOOLEAN):**
    - **Description:** A boolean value indicating whether the service is currently active (TRUE) or inactive (FALSE).

- **Purpose:** Helps track the current status of the service, used for availability management and service offerings control.

7. **start_date (TIMESTAMP):**
    - **Description:** The date and time when the service became active or was first offered.
    - **Purpose:** Helps track when the service started, useful for historical analysis and timeline management.
8. **end_date (TIMESTAMP):**
    - **Description:** The date and time when the service was discontinued or stopped being offered.
    - **Purpose:** Indicates when the service was no longer available, helping manage the service lifecycle and identify inactive services.

Table: Specialists

Purpose of the Table: The Specialists table is designed to store information about the specialists working at the company. This table helps manage specialist-specific data, including their names, specialties, contact details, experience, and availability, facilitating the assignment of specialists to services and branches.

Description of the Table: The Specialists table contains records of all specialists employed by the company. Each row represents a unique specialist identified by specialist_id. This table includes details such as the specialist's name, area of expertise, years of experience, contact information, associated branch, and availability status, along with timestamps indicating their employment period.

Column Descriptions:

1. **specialist_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each specialist. This column auto-increments for each new entry, ensuring that each specialist has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each specialist.
2. **first_name (VARCHAR(50)):**
    - **Description:** The first name of the specialist, stored as a string with a maximum length of 50 characters.
    - **Purpose:** Captures the specialist's first name, aiding in identification and personalization.
3. **last_name (VARCHAR(50)):**

- **Description:** The last name of the specialist, stored as a string with a maximum length of 50 characters.
- **Purpose:** Captures the specialist's last name, facilitating full identification.

4. **specialty (VARCHAR(100)):**
   - **Description:** The area of expertise or specialty of the specialist, stored as a string with a maximum length of 100 characters.
   - **Purpose:** Identifies the specialist's field, such as dermatology, dental care, or laser hair removal, aiding in the assignment of services.
5. **experience_years (INTEGER):**
   - **Description:** The number of years of experience the specialist has in their field.
   - **Purpose:** Indicates the specialist's experience level, useful for assigning complex tasks and promoting expertise.
6. **email (VARCHAR(100)):**
   - **Description:** The email address of the specialist, stored as a string with a maximum length of 100 characters.
   - **Purpose:** Serves as an electronic contact point, useful for internal communication and correspondence.
7. **phone_number (VARCHAR(20)):**
   - **Description:** The contact phone number of the specialist, stored as a string with a maximum length of 20 characters.
   - **Purpose:** Provides direct contact information for the specialist.
8. **branch_id (INTEGER):**
   - **Description:** A foreign key referencing the branch_id in the Branches table, linking each specialist to their assigned branch.
   - **Purpose:** Establishes the relationship between specialists and their branches, aiding in resource management and scheduling.
9. **start_date (TIMESTAMP):**
   - **Description:** The date and time when the specialist began their employment or association with the company.
   - **Purpose:** Helps track the start of the specialist's tenure, useful for historical records and employment duration analysis.
10. **end_date (TIMESTAMP):**

- **Description:** The date and time when the specialist's employment or association ended.
- **Purpose:** Indicates when the specialist left the company, aiding in employment tracking and resource management.

11. **is_available (BOOLEAN):**
    - **Description:** A boolean value indicating whether the specialist is currently available (TRUE) or unavailable (FALSE) for service assignments.
    - **Purpose:** Helps manage specialist availability, crucial for scheduling and service allocation.

Table: Availability

Purpose of the Table: The Availability table is designed to store information about the availability of specialists for client registrations. This table helps manage when specialists are available at specific branches, aiding in appointment scheduling and optimizing resource allocation.

Description of the Table: The Availability table contains records of each specialist's availability at different branches. Each row represents a specific availability slot identified by availability_id. This table includes details about which specialist is available, at which branch, on what date, and within what time range, along with timestamps indicating the period when the availability record is valid.

Column Descriptions:

1. **availability_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each availability record. This column auto-increments for each new entry, ensuring each availability slot has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each availability entry.
2. **specialist_id (INTEGER):**
    - **Description:** A foreign key referencing the specialist_id in the Specialists table, linking the availability record to a specific specialist.
    - **Purpose:** Establishes the relationship between availability slots and specialists, allowing for precise scheduling.
3. **branch_id (INTEGER):**
    - **Description:** A foreign key referencing the branch_id in the Branches table, linking the availability record to a specific branch.
    - **Purpose:** Indicates the branch where the specialist is available, helping manage location-specific scheduling.

4. **availability_date (DATE):**
    - **Description:** The date on which the specialist is available for appointments.
    - **Purpose:** Specifies the exact day of availability, crucial for scheduling and appointment management.
5. **start_time (TIME):**
    - **Description:** The start time of the availability slot, indicating when the specialist begins accepting appointments on the given date.
    - **Purpose:** Defines the starting point of the availability window, aiding in precise scheduling.
6. **end_time (TIME):**
    - **Description:** The end time of the availability slot, indicating when the specialist stops accepting appointments on the given date.
    - **Purpose:** Marks the ending point of the availability window, helping to manage appointment durations.

Table: Reservations

Purpose of the Table: The Reservations table is designed to store information about client reservations for services provided by specialists at various branches. This table helps manage reservation details, including client, specialist, service, location, and timing, facilitating appointment scheduling and tracking.

Description of the Table: The Reservations table contains records of all client reservations. Each row represents a unique reservation identified by reservation_id. The table includes details such as the client, specialist, and service involved, along with the branch, date, time, and status of the reservation. It also includes timestamps indicating the validity period of each record and a status flag to determine if the reservation is currently active.

Column Descriptions:

1. **reservation_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each reservation. This column auto-increments for each new entry, ensuring that each reservation has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each reservation record.
2. **client_id (INTEGER):**
    - **Description:** A foreign key referencing the client_id in the Clients table, linking the reservation to a specific client.

- **Purpose:** Establishes the relationship between the reservation and the client, crucial for tracking client appointments.

3. **specialist_id (INTEGER):**
    - **Description:** A foreign key referencing the specialist_id in the Specialists table, linking the reservation to a specific specialist.
    - **Purpose:** Identifies which specialist is assigned to the reservation, aiding in specialist scheduling.
4. **service_id (INTEGER):**
    - **Description:** A foreign key referencing the service_id in the Services table, linking the reservation to a specific service.
    - **Purpose:** Specifies which service is booked, helping manage service-specific scheduling and resource allocation.
5. **branch_id (INTEGER):**
    - **Description:** A foreign key referencing the branch_id in the Branches table, indicating the location of the appointment.
    - **Purpose:** Identifies the branch where the reservation will take place, useful for location-specific management.
6. **reservation_date (DATE):**
    - **Description:** The date on which the reservation is scheduled.
    - **Purpose:** Specifies when the service appointment is booked, essential for calendar management and scheduling.
7. **reservation_time (TIME):**
    - **Description:** The time at which the reservation is scheduled.
    - **Purpose:** Defines the exact time of the appointment, helping with precise scheduling and time management.
8. **reservation_status (VARCHAR(50)):**
    - **Description:** The status of the reservation, such as "Confirmed," "Cancelled," or "Pending."
    - **Purpose:** Tracks the current state of the reservation, aiding in appointment management and client communication.
9. **start_date (TIMESTAMP):**
    - **Description:** The date and time when the reservation record became active or was created.

- **Purpose:** Helps track the initiation of the reservation, useful for historical analysis and record-keeping.

10. **end_date (TIMESTAMP):**
    - **Description:** The date and time when the reservation record stopped being active, if applicable.
    - **Purpose:** Indicates when the reservation ended or was cancelled, aiding in managing past records.
11. **is_active (BOOLEAN DEFAULT TRUE):**
    - **Description:** A boolean value indicating whether the reservation is currently active (TRUE) or inactive (FALSE).
    - **Purpose:** Helps manage the reservation's status, ensuring accurate record tracking and service management.

Table: Sales

Purpose of the Table: The Sales table is designed to store information about the financial transactions related to reservations. It tracks the sales details, including the reservation linked to the sale, payment information, and any discounts applied, aiding in revenue management and financial reporting.

Description of the Table: The Sales table contains records of all sales transactions. Each row represents a unique sale identified by sale_id. The table includes details such as the associated reservation, the date of the sale, the total amount before and after discounts, and any payment linked to the transaction. This information helps manage and track the company's revenue generated from services.

Column Descriptions:

1. **sale_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each sale. This column auto-increments for each new entry, ensuring that each sale has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each sales transaction.
2. **reservation_id (INTEGER):**
    - **Description:** A foreign key referencing the reservation_id in the Reservations table, linking the sale to a specific reservation.
    - **Purpose:** Establishes the relationship between the sale and the reservation, enabling accurate tracking of service-based revenue.
3. **sale_date (DATE):**

- **Description:** The date on which the sale occurred.
- **Purpose:** Records when the transaction took place, essential for financial analysis, reporting, and revenue tracking.

4. **total_amount (NUMERIC):**
    - **Description:** The total amount of the sale before any discounts are applied.
    - **Purpose:** Reflects the initial cost of the service(s) sold, providing a baseline for revenue calculations.
5. **payment_id (INTEGER):**
    - **Description:** A foreign key that references the payment method or record associated with the sale, usually linked to a Payments table.
    - **Purpose:** Identifies how the sale was paid, aiding in financial reconciliation and payment tracking.
6. **discount_applied (NUMERIC):**
    - **Description:** The amount of discount applied to the total sale, stored as a numeric value.
    - **Purpose:** Tracks discounts given, useful for understanding pricing strategies, promotions, and net revenue impact.
7. **final_amount (NUMERIC):**
    - **Description:** The final amount paid by the client after any discounts have been applied.
    - **Purpose:** Represents the actual revenue from the sale, crucial for accurate financial reporting and analysis.

Table: Payments

Purpose of the Table: The Payments table is designed to store information about payments related to sales transactions. It records details about the payment method, the date of the payment, and its status, enabling effective tracking and management of financial transactions within the company.

Description of the Table: The Payments table contains records of all payment transactions. Each row represents a unique payment identified by payment_id. The table includes details such as the payment type, date, and status, providing an overview of how services are paid for and the current state of each payment.

Column Descriptions:

1. **payment_id (SERIAL PRIMARY KEY):**

- **Description:** A unique identifier for each payment. This column auto-increments for each new entry, ensuring that each payment has a distinct ID.
- **Purpose:** Serves as the primary key for the table, uniquely identifying each payment record.

2. **payment_type (VARCHAR(50)):**
    - **Description:** The type of payment used for the transaction, such as "Credit Card," "Cash," "Bank Transfer," or "Insurance."
    - **Purpose:** Identifies the method of payment, helping categorize and manage payment options within the company.
3. **payment_date (DATE):**
    - **Description:** The date on which the payment was made.
    - **Purpose:** Records when the payment occurred, crucial for financial tracking, reconciliation, and reporting.
4. **payment_status (VARCHAR(50)):**
    - **Description:** The status of the payment, such as "Completed," "Pending," "Failed," or "Refunded."
    - **Purpose:** Tracks the current state of the payment, aiding in financial management and ensuring the accuracy of transaction records.

Table: Feedbacks

Purpose of the Table: The Feedbacks table is designed to capture and store client feedback regarding the services they received. It helps the company assess the quality of services, specialist performance, and overall client satisfaction, which can be used to improve service offerings and address any issues.

Description of the Table: The Feedbacks table contains records of client feedback. Each row represents a unique feedback entry identified by feedback_id. The table includes details about the client, service, specialist, rating, comments, and the date the feedback was given. This information is essential for evaluating and enhancing service quality and client experience.

Column Descriptions:

1. **feedback_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each feedback entry. This column auto-increments for each new entry, ensuring that each feedback record has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each feedback record.

2. **client_id (INTEGER):**
    - **Description:** A foreign key referencing the client_id in the Clients table, linking the feedback to a specific client.
    - **Purpose:** Establishes the relationship between the feedback and the client, helping identify which client provided the feedback.
3. **service_id (INTEGER):**
    - **Description:** A foreign key referencing the service_id in the Services table, linking the feedback to a specific service.
    - **Purpose:** Indicates which service the feedback is related to, providing context for the feedback provided.
4. **specialist_id (INTEGER):**
    - **Description:** A foreign key referencing the specialist_id in the Specialists table, linking the feedback to a specific specialist.
    - **Purpose:** Identifies which specialist provided the service, allowing for performance evaluation.
5. **rating (INTEGER CHECK (rating BETWEEN 1 AND 5)):**
    - **Description:** A numeric rating provided by the client, ranging from 1 to 5, where 1 typically represents poor and 5 represents excellent.
    - **Purpose:** Provides a quantitative measure of client satisfaction, facilitating performance assessment and improvement.
6. **feedback_comments (TEXT):**
    - **Description:** Textual comments provided by the client, offering detailed feedback and observations about their experience.
    - **Purpose:** Allows clients to elaborate on their experience, providing qualitative insights that complement the rating.
7. **feedback_date (DATE):**
    - **Description:** The date on which the feedback was given.
    - **Purpose:** Records when the feedback was submitted, useful for tracking feedback trends over time and managing responses.

Table: Promotions

Purpose of the Table: The Promotions table is designed to store information about promotional offers provided by the company. It helps manage and track the details of promotions, including their names, discount percentages, validity periods, and current status. This information is essential for managing marketing campaigns and applying discounts to services.

Description of the Table: The Promotions table contains records of all promotional offers. Each row represents a unique promotion identified by promotion_id. The table includes details such as the promotion's name, discount percentage, validity period, and status. This data supports effective promotion management and analysis.

Column Descriptions:

1. **promotion_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each promotion. This column auto-increments for each new entry, ensuring that each promotion has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each promotional offer.
2. **promotion_name (VARCHAR(100)):**
    - **Description:** The name of the promotion, stored as a string with a maximum length of 100 characters.
    - **Purpose:** Provides a descriptive name for the promotion, making it easier to reference and identify in marketing materials and reports.
3. **discount_percentage (NUMERIC):**
    - **Description:** The percentage discount applied during the promotion, stored as a numeric value.
    - **Purpose:** Specifies the amount of discount given, essential for calculating the reduced price of services.
4. **start_date (DATE):**
    - **Description:** The date when the promotion starts being valid.
    - **Purpose:** Marks the beginning of the promotion period, helping to manage the timing and availability of the promotional offer.
5. **end_date (DATE):**
    - **Description:** The date when the promotion ends or stops being valid.
    - **Purpose:** Indicates when the promotion period concludes, aiding in tracking the duration of the promotional offer.
6. **is_active (BOOLEAN):**
    - **Description:** A boolean value indicating whether the promotion is currently active (TRUE) or inactive (FALSE).
    - **Purpose:** Helps manage the promotion’s status, ensuring accurate application of the offer and aiding in promotional campaign management.

Table: Insurance_Providers

Purpose of the Table: The Insurance_Providers table is designed to store information about insurance providers that are affiliated with the company. It helps manage and track details about each insurance provider, including their contact information, address, and the validity period of their affiliation. This information is essential for managing insurance-related services and communications.

Description of the Table: The Insurance_Providers table contains records of insurance providers. Each row represents a unique provider identified by provider_id. The table includes details such as the provider's name, contact information, address, and the period during which the provider is affiliated with the company. This information supports effective management of insurance provider relationships and services.

Column Descriptions:

1. **provider_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each insurance provider. This column auto-increments for each new entry, ensuring that each provider has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each insurance provider record.
2. **provider_name (VARCHAR(100)):**
    - **Description:** The name of the insurance provider, stored as a string with a maximum length of 100 characters.
    - **Purpose:** Provides the name of the provider, making it easier to reference and identify in records and communications.
3. **contact_number (VARCHAR(20)):**
    - **Description:** The contact phone number of the insurance provider, stored as a string with a maximum length of 20 characters.
    - **Purpose:** Provides a phone number for contacting the insurance provider, essential for communication and support.
4. **email (VARCHAR(100)):**
    - **Description:** The email address of the insurance provider, stored as a string with a maximum length of 100 characters.
    - **Purpose:** Provides an email address for digital communication and correspondence with the insurance provider.
5. **address (VARCHAR(255)):**
    - **Description:** The physical address of the insurance provider, stored as a string with a maximum length of 255 characters.

- **Purpose:** Provides the location of the insurance provider's office, useful for correspondence and service management.

6. **city_id (INTEGER):**
    - **Description:** A foreign key referencing the city_id in the Cities table, linking the provider's address to a specific city.
    - **Purpose:** Identifies the city where the insurance provider is located, helping manage location-specific information.
7. **start_date (TIMESTAMP):**
    - **Description:** The date and time when the insurance provider began their affiliation with the company.
    - **Purpose:** Marks the start of the provider's relationship with the company, aiding in tracking and managing the affiliation.
8. **end_date (TIMESTAMP):**
    - **Description:** The date and time when the insurance provider's affiliation ended or was terminated.
    - **Purpose:** Indicates the end of the provider's relationship with the company, useful for managing historical records and current affiliations.

Table: Insurance_Policies

Purpose of the Table: The Insurance_Policies table is designed to store information about insurance policies provided by insurance providers. It helps manage and track details about each policy, including the associated provider, policy name, coverage details, and validity period. This information is crucial for managing insurance offerings and client coverage options.

Description of the Table: The Insurance_Policies table contains records of insurance policies. Each row represents a unique policy identified by policy_id. The table includes details such as the policy's provider, coverage amount, type, and validity dates. This data supports effective management of insurance policies and client coverage.

Column Descriptions:

1. **policy_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each insurance policy. This column auto-increments for each new entry, ensuring that each policy has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each insurance policy record.
2. **provider_id (INTEGER):**

- **Description:** A foreign key referencing the provider_id in the Insurance_Providers table, linking the policy to a specific insurance provider.
- **Purpose:** Establishes the relationship between the policy and the provider, identifying which provider offers the policy.

3. **policy_name (VARCHAR(100)):**
    - **Description:** The name of the insurance policy, stored as a string with a maximum length of 100 characters.
    - **Purpose:** Provides a descriptive name for the policy, making it easier to reference and identify in records and client communications.
4. **coverage_amount (NUMERIC):**
    - **Description:** The amount of coverage provided by the policy, stored as a numeric value.
    - **Purpose:** Specifies the financial coverage offered by the policy, important for determining the extent of client protection.
5. **policy_type (VARCHAR(50)):**
    - **Description:** The type of insurance policy, such as "Health," "Dental," or "Life," stored as a string with a maximum length of 50 characters.
    - **Purpose:** Categorizes the policy based on its type, aiding in managing and differentiating various insurance products.
6. **expiry_date (DATE):**
    - **Description:** The date on which the policy expires or is no longer valid.
    - **Purpose:** Indicates the end of the policy's validity period, helping manage policy renewals and client coverage status.
7. **start_date (TIMESTAMP):**
    - **Description:** The date and time when the policy began being effective.
    - **Purpose:** Marks the start of the policy's validity, useful for tracking the policy's coverage period.
8. **end_date (TIMESTAMP):**
    - **Description:** The date and time when the policy ended or was terminated.
    - **Purpose:** Indicates the end of the policy's coverage period, important for managing historical records and policy transitions.

Table: Client_Insurance

Purpose of the Table: The Client_Insurance table is designed to store information about insurance policies that clients have taken out. It manages the relationship between clients and their insurance policies, including details about the policy number, coverage percentage, and the validity of the insurance. This information is essential for tracking and managing client insurance coverage and ensuring that clients receive the appropriate benefits.

Description of the Table: The Client_Insurance table contains records of insurance policies associated with clients. Each row represents a unique client-insurance relationship identified by client_insurance_id. The table includes details such as the client, the policy they hold, the policy number, coverage details, and the validity of the insurance. This data supports effective management of client insurance and coverage records.

Column Descriptions:

1. **client_insurance_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each client-insurance record. This column auto-increments for each new entry, ensuring that each record has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each client-insurance relationship.
2. **client_id (INTEGER):**
    - **Description:** A foreign key referencing the client_id in the Clients table, linking the insurance record to a specific client.
    - **Purpose:** Establishes the relationship between the client and their insurance policy, identifying which client holds the policy.
3. **policy_id (INTEGER):**
    - **Description:** A foreign key referencing the policy_id in the Insurance_Policies table, linking the insurance record to a specific insurance policy.
    - **Purpose:** Connects the client's insurance record to the specific policy they have, facilitating management of policy details.
4. **policy_number (VARCHAR(100)):**
    - **Description:** The unique number assigned to the insurance policy, stored as a string with a maximum length of 100 characters.
    - **Purpose:** Provides a unique identifier for the policy, making it easier to reference and manage the policy within records.
5. **coverage_percentage (NUMERIC):**
    - **Description:** The percentage of coverage provided by the insurance policy, stored as a numeric value.

- **Purpose:** Indicates the proportion of the service or cost covered by the insurance, important for calculating client benefits and claims.

6. **start_date (TIMESTAMP):**
    - **Description:** The date and time when the insurance coverage began.
    - **Purpose:** Marks the start of the client's insurance coverage period, useful for managing the validity and application of coverage.
7. **end_date (TIMESTAMP):**
    - **Description:** The date and time when the insurance coverage ended or was terminated.
    - **Purpose:** Indicates the end of the client's insurance coverage period, important for managing coverage transitions and expirations.
8. **is_active (BOOLEAN):**
    - **Description:** A boolean value indicating whether the insurance coverage is currently active (TRUE) or inactive (FALSE).
    - **Purpose:** Helps manage the status of the client's insurance coverage, ensuring accurate application of benefits and handling of claims.

Table: Insurance_Claims

Purpose of the Table: The Insurance_Claims table is designed to store information about insurance claims submitted by clients. It helps manage and track details of each claim, including the associated policy, client, reservation, claim amount, status, and relevant dates. This information is essential for processing claims, assessing coverage, and ensuring that claims are handled efficiently.

Description of the Table: The Insurance_Claims table contains records of insurance claims made by clients. Each row represents a unique claim identified by claim_id. The table includes details such as the policy and client involved, the claim amount, status, submission and approval dates, and any additional remarks. This data supports effective management and processing of insurance claims.

Column Descriptions:

1. **claim_id (SERIAL PRIMARY KEY):**
    - **Description:** A unique identifier for each insurance claim. This column auto-increments for each new entry, ensuring that each claim has a distinct ID.
    - **Purpose:** Serves as the primary key for the table, uniquely identifying each insurance claim record.
2. **policy_id (INTEGER):**

- **Description:** A foreign key referencing the policy_id in the Insurance_Policies table, linking the claim to a specific insurance policy.
- **Purpose:** Connects the claim to the insurance policy under which it is filed, facilitating policy-related processing and tracking.

3. **client_id (INTEGER):**
   - **Description:** A foreign key referencing the client_id in the Clients table, linking the claim to a specific client.
   - **Purpose:** Identifies the client who has submitted the claim, providing context for the claim and supporting client-specific processing.
4. **reservation_id (INTEGER):**
   - **Description:** A foreign key referencing the reservation_id in the Reservations table, linking the claim to a specific reservation.
   - **Purpose:** Connects the claim to a particular reservation, useful for verifying the claim's details and processing it in relation to the reservation.
5. **claim_amount (NUMERIC):**
   - **Description:** The amount of money claimed by the client, stored as a numeric value.
   - **Purpose:** Specifies the financial amount being claimed, important for processing and approving the claim.
6. **claim_status (VARCHAR(50)):**
   - **Description:** The current status of the claim, such as "Pending," "Approved," "Rejected," or "Settled," stored as a string with a maximum length of 50 characters.
   - **Purpose:** Indicates the progress and outcome of the claim, aiding in claim management and tracking.
7. **submission_date (DATE):**
   - **Description:** The date on which the claim was submitted by the client.
   - **Purpose:** Records when the claim was filed, important for tracking the claim's processing timeline and deadlines.
8. **approval_date (DATE):**
   - **Description:** The date on which the claim was approved by the insurance provider.
   - **Purpose:** Marks when the claim was officially approved, useful for managing claim approvals and processing.
9. **remarks (TEXT):**

- **Description:** Additional comments or notes related to the claim, stored as text.
- **Purpose:** Provides space for detailed notes or explanations about the claim, aiding in comprehensive claim processing and communication.

10. **start_date (TIMESTAMP):**
    - **Description:** The date and time when the claim process began.
    - **Purpose:** Marks the initiation of the claim, useful for tracking the claim's progress and managing processing timelines.
11. **end_date (TIMESTAMP):**
    - **Description:** The date and time when the claim process ended or was resolved.
    - **Purpose:** Indicates the completion of the claim process, important for managing claim closures and record-keeping.

Foreign Key Constraints Documentation

1. Branches Table:

- **fk_branches_city_id**
  - **Description:** Links the city_id column in the Branches table to the city_id column in the Cities table.
  - **Purpose:** Ensures that each branch is associated with a valid city.

2. Clients Table:

- **fk_clients_city_id**
  - **Description:** Links the city_id column in the Clients table to the city_id column in the Cities table.
  - **Purpose:** Ensures that each client is associated with a valid city.
- **fk_clients_branch_id**
  - **Description:** Links the branch_id column in the Clients table to the branch_id column in the Branches table.
  - **Purpose:** Ensures that each client is associated with a valid branch.

3. Services Table:

- **fk_services_category_id**

    - **Description:** Links the category_id column in the Services table to the category_id column in the Categories table.
    - **Purpose:** Ensures that each service belongs to a valid category.

4. Specialists Table:

- **fk_specialists_branch_id**
    - **Description:** Links the branch_id column in the Specialists table to the branch_id column in the Branches table.
    - **Purpose:** Ensures that each specialist is associated with a valid branch.

5. Availability Table:

- **fk_availability_specialist_id**
    - **Description:** Links the specialist_id column in the Availability table to the specialist_id column in the Specialists table.
    - **Purpose:** Ensures that the availability records are associated with valid specialists.
- **fk_availability_branch_id**
    - **Description:** Links the branch_id column in the Availability table to the branch_id column in the Branches table.
    - **Purpose:** Ensures that the availability records are associated with valid branches.

6. Reservations Table:

- **fk_reservations_client_id**
    - **Description:** Links the client_id column in the Reservations table to the client_id column in the Clients table.
    - **Purpose:** Ensures that each reservation is made by a valid client.
- **fk_reservations_specialist_id**
    - **Description:** Links the specialist_id column in the Reservations table to the specialist_id column in the Specialists table.
    - **Purpose:** Ensures that each reservation is associated with a valid specialist.
- **fk_reservations_service_id**
    - **Description:** Links the service_id column in the Reservations table to the service_id column in the Services table.
    - **Purpose:** Ensures that each reservation includes a valid service.
- **fk_reservations_branch_id**

- **Description:** Links the branch_id column in the Reservations table to the branch_id column in the Branches table.
- **Purpose:** Ensures that each reservation is associated with a valid branch.

7. Sales Table:

- **fk_sales_reservation_id**
  - **Description:** Links the reservation_id column in the Sales table to the reservation_id column in the Reservations table.
  - **Purpose:** Ensures that each sale is linked to a valid reservation.
- **fk_sales_payment_id**
- **fk_sales_payment_id**
  - **Description:** Links the payment_id column in the Sales table to the payment_id column in the Payments table.
  - **Purpose:** Ensures that each sale is associated with a valid payment record.

8. Feedbacks Table:

- **fk_feedbacks_client_id**
  - **Description:** Links the client_id column in the Feedbacks table to the client_id column in the Clients table.
  - **Purpose:** Ensures that each feedback record is associated with a valid client.
- **fk_feedbacks_service_id**
  - **Description:** Links the service_id column in the Feedbacks table to the service_id column in the Services table.
  - **Purpose:** Ensures that each feedback record pertains to a valid service.
- **fk_feedbacks_specialist_id**
  - **Description:** Links the specialist_id column in the Feedbacks table to the specialist_id column in the Specialists table.
  - **Purpose:** Ensures that each feedback record is associated with a valid specialist.

9. Insurance_Providers Table:

- **fk_insurance_providers_city_id**
  - **Description:** Links the city_id column in the Insurance_Providers table to the city_id column in the Cities table.
  - **Purpose:** Ensures that each insurance provider is associated with a valid city.

10. Insurance_Policies Table:

- **fk_insurance_policies_provider_id**
    - **Description:** Links the provider_id column in the Insurance_Policies table to the provider_id column in the Insurance_Providers table.
    - **Purpose:** Ensures that each insurance policy is linked to a valid insurance provider.

13. Client_Insurance Table:

- **fk_client_insurance_client_id**
    - **Description:** Links the client_id column in the Client_Insurance table to the client_id column in the Clients table.
    - **Purpose:** Ensures that each client-insurance record is associated with a valid client.
- **fk_client_insurance_policy_id**
    - **Description:** Links the policy_id column in the Client_Insurance table to the policy_id column in the Insurance_Policies table.
    - **Purpose:** Ensures that each client-insurance record pertains to a valid insurance policy.

14. Insurance_Claims Table:

- **fk_insurance_claims_policy_id**
    - **Description:** Links the policy_id column in the Insurance_Claims table to the policy_id column in the Insurance_Policies table.
    - **Purpose:** Ensures that each insurance claim is associated with a valid insurance policy.
- **fk_insurance_claims_client_id**
    - **Description:** Links the client_id column in the Insurance_Claims table to the client_id column in the Clients table.
    - **Purpose:** Ensures that each insurance claim is submitted by a valid client.
- **fk_insurance_claims_reservation_id**
    - **Description:** Links the reservation_id column in the Insurance_Claims table to the reservation_id column in the Reservations table.
    - **Purpose:** Ensures that each insurance claim is related to a valid reservation.

Documentation for Table Updates and Row Additions

1. Branches Table

Updating Information:

- **branch_name:** Update when the name of the branch changes.
- **address:** Update when the address of the branch changes.
- **phone_number:** Update when the phone number of the branch changes.
- **email:** Update when the email address of the branch changes.
- **branch_status:** Update when the status (active/inactive) of the branch changes.

Adding a Row:

- **When to Add a Row:** Add a new row when a new branch is established.

2. Clients Table

Updating Information:

- **first_name:** Add a new row.
- **last_name:** Add a new row.
- **date_of_birth:** Add a new row.
- **email:** Update when the client's email address changes.
- **phone_number:** Update when the client's phone number changes.
- **address:** Update if the client's address changes.
- **city_id:** Update if the client moves to a different city.
- **branch_id:** Update if the client’s associated branch changes.
- **gender:** Add a new row.
- **start_date:** Add a new row.
- **end_date:** Add a new row.

3. Services Table

Updating Information:

- **service_name:** Update/ Add a new row.
- **price:** Add a new row.
- **duration:** Update/ Add a new row.
- **is_active:** Update to activate or deactivate the service.
- **start_date:** Add a new row.

- **end_date:** Update if the service stops being offered from a different date.

4. Specialists Table

Updating Information:

- **specialist_id:** Add a new row.
- **first_name:** Add a new row.
- **last_name:** Add a new row.
- **specialty:** Update/ Add a new row.
- **experience_years:** Update if there is a change in the specialist's years of experience.
- **email:** Update when the specialist's email address changes.
- **phone_number:** Update when the specialist's phone number changes.
- **branch_id:** Update/ Add a new row.
- **start_date:** Add a new row.
- **end_date:** Update if the specialist's end date at the branch changes or if they leave the branch.
- **is_available:** Update to reflect the specialist's availability status.

5. Availability Table

Updating Information:

- **availability_date:** Update if there is a change in the availability date of the specialist.
- **start_time:** Update if there is a change in the start time of the specialist's availability.
- **end_time:** Update if there is a change in the end time of the specialist's availability.
- **branch_id:** Update/ Add a new row.
- **specialist_id:** Update.

6. Reservations Table

Updating Information:

- **reservation_date:** Update if the date of the reservation changes.
- **reservation_time:** Update if the time of the reservation changes.
- **reservation_status:** Update if the status of the reservation changes.
- **start_date:** Update if the start date of the reservation record changes.

- **end_date:** Update if the end date of the reservation record changes (for archiving purposes).

Adding a Row:

- **When to Add a Row:** Add a new row when a new reservation is made.

7. Sales Table

Updating Information:

- **sale_date:** Update if the date of the sale changes.
- **total_amount:** Update if the total amount of the sale changes.
- **discount_applied:** Update if there is a change in the discount applied to the sale.
- **final_amount:** Update if the final amount of the sale changes after adjustments.

Adding a Row:

- **When to Add a Row:** Add a new row for each new sale transaction.

8. Payments Table

Updating Information:

- **payment_type:** Update if the type of payment changes.
- **payment_date:** Update if the date of payment changes.
- **payment_status:** Update if the status of the payment changes.

Adding a Row:

- **When to Add a Row:** Add a new row for each new payment record.

9. Feedbacks Table

Updating Information:

- **rating:** Update if the rating given by the client changes.
- **feedback_comments:** Update if there are changes to the comments provided in the feedback.
- **feedback_date:** Update if the date of the feedback changes.

Adding a Row:

- **When to Add a Row:** Add a new row for each new feedback received.

10. Insurance_Providers Table

Updating Information:

- **provider_name:** Update if the name of the insurance provider changes.
- **contact_number:** Update if the contact number of the provider changes.
- **email:** Update if the email address of the provider changes.
- **address:** Update if the address of the provider changes.
- **city_id:** Update if the provider moves to a different city.

Adding a Row:

- **When to Add a Row:** Add a new row when a new insurance provider is added.

11. Insurance_Policies Table

Updating Information:

- **policy_name:** Update if the name of the policy changes.
- **coverage_amount:** Update if the coverage amount changes.
- **policy_type:** Update if the type of policy changes.
- **expiry_date:** Update if the expiry date of the policy changes.
- **start_date:** Update if the start date of the policy changes.
- **end_date:** Update if the end date of the policy changes.

Adding a Row:

- **When to Add a Row:** Add a new row when a new insurance policy is introduced.

12. Client_Insurance Table

Updating Information:

- **policy_number:** Update if the policy number changes.
- **coverage_percentage:** Update if the coverage percentage changes.
- **start_date:** Update if the start date of the insurance coverage changes.
- **end_date:** Update if the end date of the insurance coverage changes.
- **is_active:** Update to reflect if the insurance coverage is active or not.

Adding a Row:

- **When to Add a Row:** Add a new row when a client acquires a new insurance policy.

13. Insurance_Claims Table

Updating Information:

- **claim_amount:** Update if the claimed amount changes.

- **claim_status:** Update if the status of the claim changes.
- **submission_date:** Update if the submission date of the claim changes.
- **approval_date:** Update if the approval date of the claim changes.
- **remarks:** Update if additional remarks or comments are added or changed.

Adding a Row:

- **When to Add a Row:** Add a new row when a new insurance claim is filed.

NOTE: The update and add row functions are relevant considering the archival period of the organization.