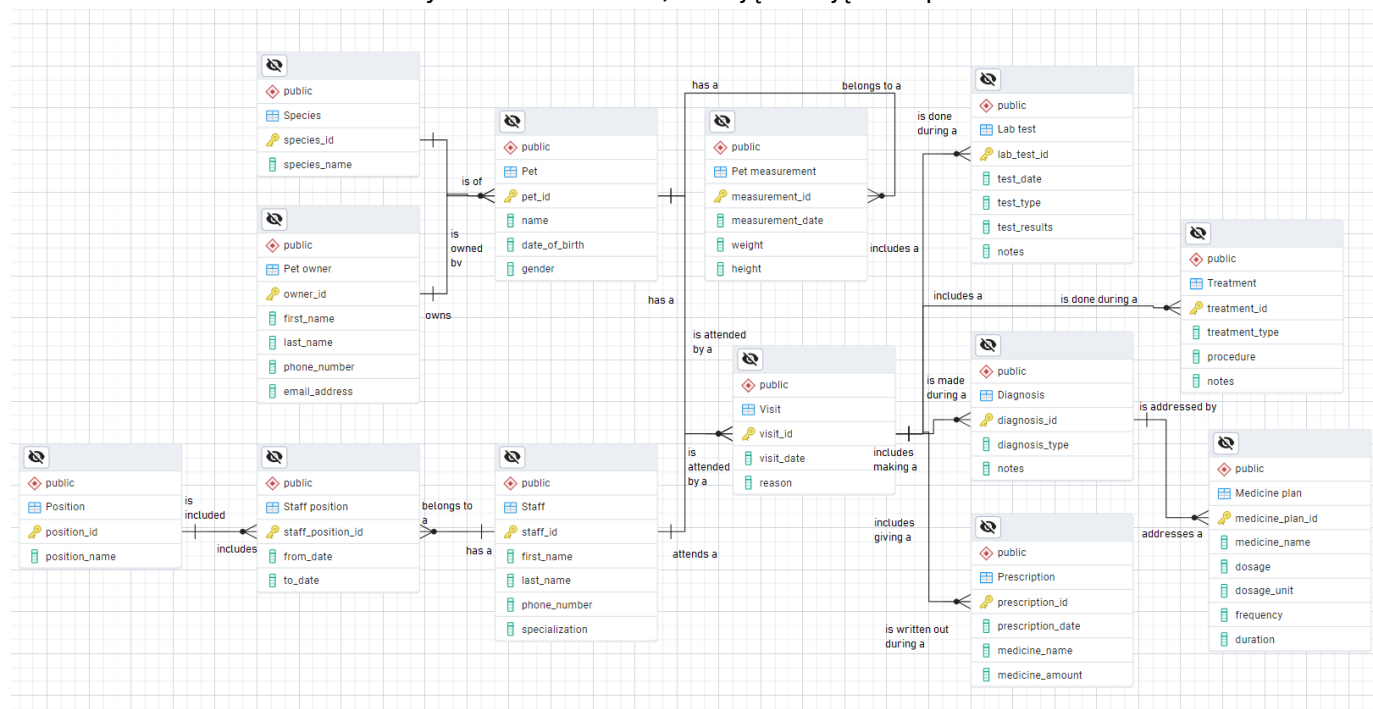


Sprawozdanie – Relacyjne systemy zarządzania bazami danych - Etap I

Poprawiony Graf ERD

Poprawiono graf ERD. Dodano historyczność pomiarów zwierząt poprzez dodanie encję Pet measurement. Dodano też historyczność stanowisk, dodając encję Staff position.



Baza danych (i uzasadnienia istotnych decyzji)

Stworzono bazę danych z 13 tabelami.

Stworzono trzy tabele dotyczące pracowników i ich stanowisk: Staff (pracownicy), Staff position (pozycja pracownika) oraz Position (pozycja / stanowisko). Tabela Position przechowuje tytuły stanowisk, takie jak „Weterynarz”, „Sprzątac” – stworzoną tą tabelę aby ujednolicić nazwy stanowisk w klinice i ich nie powtarzać. Staff position reprezentuje zatrudnienie pracownika i umożliwia przechowywanie historycznych danych. Przechowuje ona datę rozpoczęcia i zakończenia pracy na danym stanowisku. Data zakończenia pracy jest atrybutem nieobowiązkowym. Tabela ta ma związek N-1 z Staff, ponieważ założono, że dany pracownik może np. zmienić stanowisko, dostać awans, zrezygnować z pracy w klinice a następnie do niej wrócić.

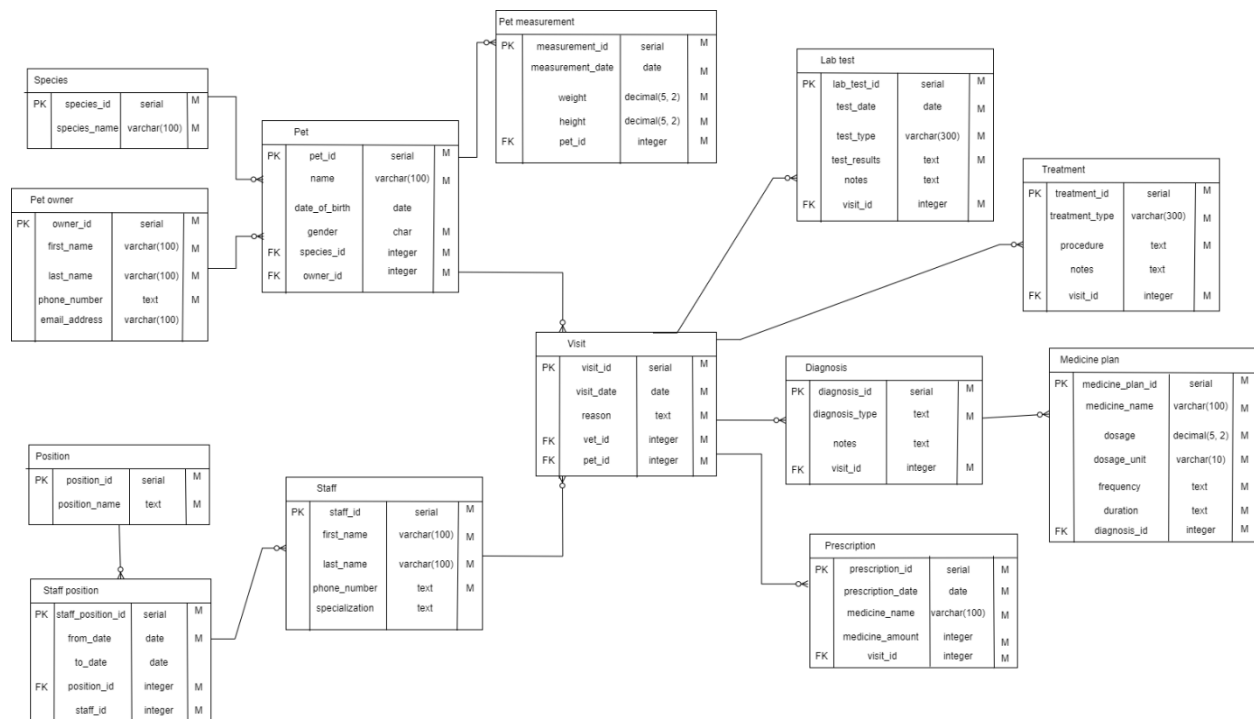
Tabela Pet measurement reprezentuje pomiary wzrostu i wagi zwierzęcia. Wartości te mogą się zmieniać z czasem, szczególnie w okresie dojrzewania zwierzęcia, więc jedno zwierzę może mieć

przypisanych wiele pomiarów, które np. odbyły się co jakiś czas. Tym samym, tabela ta zapewnia możliwość przechowywania historycznych danych.

Podczas wizyty (Visit) może się odbyć wiele badań (Lab test), wiele procedur medycznych (Treatment), może być postawionych wiele diagnoz (Diagnosis) i być wypisane wiele recept (Prescription), dlatego też wszystkie wymienione tabele mają relacje 1-N z tabelą Visit. Wszystkie pola „notes” w powyższych tabelach są nieobowiązkowe, ponieważ reprezentują pole na dodatkowe uwagi weterynarzy (których akurat może nie być).

Data urodzenia zwierzęcia (date_of_birth) jest polem nieobowiązkowym, ponieważ może być nieznana (jeśli np. zwierzę zostało znalezione lub adoptowane). Pola frequency i duration w Medicine plan (reprezentującym plan leczenia zwierzęcia) są typu text, aby dać swobodę w wpisywaniu / określaniu częstotliwości i okresu brania leków (weterynarz może np. chcieć określić okres jako „aż będzie poprawa” zamiast podania liczby dni).

Schemat logiczny



Użytkownicy i perspektywy

Stworzono 14 użytkowników: 5 pracowników-weterynarzy, 4 (innych) pracowników i 5 właścicieli zwierząt.

Stworzono widoki dla weterynarzy pozwalające im na wyświetlenie w czytelny sposób swoich wizyt, procedur medycznych, testów, diagnoz, planów lekowych, recept i pomiarów swoich zwierząt-

pacjentów. Posiadają oni też dostęp do widoków ukazujących dane swoich zwierząt-pacjentów i ich właścicieli. Mogą też zobaczyć swoje dane i dane na temat swojego zatrudnienia.

Użytkownicy nie mają dostępu bezpośrednio do tabel (np. do tabeli visit):

```
1 set role amazurczak;
2 select * from visit;
```

Data Output Messages Notifications

ERROR: permission denied for table visit

SQL state: 42501

Lecz mają dostęp do swoich widoków, które w czytelny sposób przedstawiają im informacje na temat np. ich wizyt:

```
1 set role amazurczak;
2 select * from vet_amazurczak_visits;
```

Data Output Messages Notifications

	visit_id integer	visit_date timestamp without time zone	reason text	pet_name character varying (100)	pet_owner text
1	7	2021-07-22 11:00:00	Rutynowa kontrola	Laweta	Mariusz Lesner
2	3	2019-11-24 15:00:00	Zmęczenie i dziwne zachowanie	Laweta	Mariusz Lesner
3	5	2020-12-12 17:00:00	Igły jeża wbite w psa	Dakota	Cezary Wysocki

Widoki dla weterynarzy zostały przygotowane z myślą o przystępnym dostępie do danych. Dlatego też, np. przy wyświetlaniu widoków na temat procedur medycznych wykonanych przez weterynarza, wyświetla się również imię pacjenta-zwierzęcia i data wizyty, podczas której została ona wykonana.

```
1 set role jsierakowski;
2 select * from vet_jsierakowski_treatments;
```

Data Output Messages Notifications

	treatment_id integer	visit_date timestamp without time zone	pet_name character varying (100)	treatment_type character varying (300)	procedure text	notes text
1	1	2019-04-20 09:00:00	Napoleon	Szczepienie na wściekliznę	Podano psu szczepionkę	[null]
2	2	2019-04-20 09:00:00	Napoleon	Podanie tabletki	Podano psu tabletkę na kleszcze zakupioną podczas wizy...	[null]

Weterynarze mają też dostęp do widoków wyświetlających testy przez nich wykonane:

```
1 set role switowska;
2 select * from vet_switowska_lab_tests;
```

Data Output Messages Notifications

	lab_test_id integer	pet_name character varying (100)	test_date timestamp without time zone	test_type character varying (300)	test_results text	notes text
1	4	Lili	2021-07-20 00:00:00	RTG	Złamanie lewej przedniej kończyny w trzech miejscac...	[null]
2	5	Lili	2021-07-20 00:00:00	Hemoglobina	74.32 - znacznie poniżej normy	[null]

Widoki udostępniają też weterynarzom wgląd w pomiary zwierząt którymi się zajmowali/zajmują:

```

1 set role jmrotek;
2 select * from vet_jmrotek_measurements;

```

Data Output Messages Notifications

	measurement_id integer	pet_name character varying (100)	measurement_date date	weight numeric (5,2)	height numeric (5,2)
1	4	Lea	2020-07-23	0.46	10.40

```

1 set role ajanowicz;
2 select * from vet_ajanowicz_pets;

```

Data Output Messages Notifications

	pet_id integer	pet_name character varying (100)	species character varying (100)	date_of_birth date	gender "char"	latest_weight numeric (5,2)	latest_height numeric (5,2)	pet_owner text
1	2	Klatka	Kaczka	2019-01-15	M	3.60	34.70	Patryk Galus

Dodano 4 użytkowników-pracowników: 2 recepcjonistów, 1 technika sprzętowego i 1 sprzątacza. Recepcjoniści mają wgląd we wszystkie wizyty, informacje na temat pracowników i ich zatrudnienia, zwierząt zarejestrowanych w klinice oraz ich właścicieli. Technik sprzętowy ma wgląd w swoje dane i w wykonane badania, bez ich wyników. Sprzątacze ma jedynie wgląd w swoje dane.

Widok technika sprzętowego na wykonane testy:

```

1 set role jbiskupska;
2 select * from tech_tests;

```

Data Output Messages Notifications

	lab_test_id integer	test_date timestamp without time zone	test_type character varying (300)
1	1	2019-11-24 00:00:00	FT4
2	2	2020-12-12 00:00:00	Leukocyty
3	3	2023-04-16 00:00:00	USG
4	4	2021-07-20 00:00:00	RTG
5	5	2021-07-20 00:00:00	Hemoglobina

Widok recepcjonisty na wizyty:

```

1  set role jopolska;
2  select * from recept_visits;

```

visit_id	visit_date	reason	pet_name	pet_owner
integer	timestamp without time zone	text	character varying (100)	text
1	2019-04-20 09:00:00	Szczepienie psa	Napoleon	Milena Kortaba
2	2019-07-11 12:00:00	Nagle osłabienie	Klatka	Patryk Galus
3	2019-11-24 15:00:00	Zmęczenie i dziwne zachowanie	Laweta	Mariusz Lesner
4	2020-07-23 14:00:00	Rutynowa kontrola	Lea	Alicja Pawliczak
5	2020-12-12 17:00:00	Igły jeża wbite w psa	Dakota	Cezary Wysocki
6	2021-07-20 09:00:00	Potrącenie przez samochód	Lili	Mariusz Lesner
7	2021-07-22 11:00:00	Rutynowa kontrola	Laweta	Mariusz Lesner
8	2021-07-26 09:00:00	Niepokojące zachowanie po operacji	Lili	Mariusz Lesner
9	2022-11-13 13:00:00	Podejrzenie urazu oka	Klatka	Patryk Galus
10	2023-04-16 12:00:00	Podejrzenie ciąży psa	Dakota	Cezary Wysocki

Wgląd sprzątacza w swoje dane:

```

1  set role sdanielczyk;
2  select * from clean_sdanielczyk_staff;

```

staff_id	name	phone_number	position	specialization
integer	text	text	text	text
2	Sebastian Danielczyk	836-213-132	Sprzątacza	[null]

Właściciele zwierząt mają wgląd dane swoje i swoich zwierząt. Mają też wgląd w pomiary, wizyty, weterynarzy, diagnozy, testy, procedury medyczne, recepty i plany lekowe swoich zwierząt.

Przykładowe widoki dla właścicieli zwierząt:

```

1  set role apawliczak;
2  select * from owner_apawliczak_pets;

```

pet_id	pet_name	species	date_of_birth	gender	latest_weight	latest_height
integer	character varying (100)	character varying (100)	date	"char"	numeric (5,2)	numeric (5,2)
4	Lea	Żółw	2016-06-25	F	0.46	10.40

```

1  set role cwysocki;
2  select * from owner_cwysocki_prescriptions;

```

prescription_id	pet_name	prescription_date	medicine_name	medicine_amount	vet
integer	character varying (100)	date	character varying (100)	integer	text
7	Dakota	2020-12-12	Amoksylicyna	1	Anita Mazurczak

```
1 set role mkortaba;
2 select * from owner_mkortaba_treatments;
```

Data Output Messages Notifications

	treatment_id integer	visit_date timestamp without time zone	pet_name character varying (100)	treatment_type character varying (300)	procedure text	notes text	vet text
1	1	2019-04-20 09:00:00	Napoleon	Szczepienie na wściekliznę	Podano psu szczepionkę	[null]	Jakub Sierakowski
2	2	2019-04-20 09:00:00	Napoleon	Podanie tabletki	Podano psu tabletkę na kleszcze zakupioną podczas wizy...	[null]	Jakub Sierakowski

```
1 set role mlesner;
2 select * from owner_mlesner_visits;
```

Data Output Messages Notifications

	visit_id integer	visit_date timestamp without time zone	reason text	pet_name character varying (100)	vet text
1	3	2019-11-24 15:00:00	Zmęczenie i dziwne zachowanie	Laweta	Anita Mazurczak
2	6	2021-07-20 09:00:00	Potrącenie przez samochód	Lili	Sylvia Witowska
3	7	2021-07-22 11:00:00	Rutynowa kontrola	Laweta	Anita Mazurczak
4	8	2021-07-26 09:00:00	Niepokojące zachowanie po operacji	Lili	Jakub Sierakowski

```
1 set role pgalus;
2 select * from owner_pgalus_diagnosis;
```

Data Output Messages Notifications

	diagnosis_id integer	visit_date timestamp without time zone	pet_name character varying (100)	diagnosis_type text	notes text	vet text
1	1	2022-11-13 13:00:00	Kłatka	Ślepota	Kaczka ma lekką ślepotę lewego oka	Artur Janowicz

Każdy użytkownik-weterynarz ma przypisane 11 widoków. Każdy użytkownik-właściciel ma dostęp do 10 widoków. Sumarycznie, stworzono 15 widoków dla pracowników (oprócz widoków stworzonych dla weterynarzy). W zrzutach ekranu powyżej pokazano niektóre z tych widoków, dokumentując przy okazji poprawny dostęp wszystkich użytkowników.

Zapytania SQL

Zapytanie 1

Pierwsze zapytanie dotyczyło ostatnich procedur medycznych (treatment) zwierząt zarejestrowanych w klinice i dat ich wykonania (dat wizyt podczas których zostały wykonane). Nie zwrócono danych dla zwierząt które nigdy nie miały wykonanej żadnej procedury medycznej.

```

1  -- Query 1
2
3  v SELECT
4      pet.pet_name,
5      latest_treatment.treatment_type,
6      latest_treatment.visit_date
7  FROM
8      pet
9  LEFT JOIN (
10     SELECT
11         visit.pet_id,
12         t.treatment_type,
13         visit.visit_date,
14         ROW_NUMBER() OVER (PARTITION BY visit.pet_id ORDER BY visit.visit_date DESC) AS rn
15     FROM
16         visit
17     JOIN treatment t ON visit.visit_id = t.visit_id
18 ) AS latest_treatment ON pet.pet_id = latest_treatment.pet_id AND latest_treatment.rn = 1
19 WHERE latest_treatment.treatment_type is not null
20 ORDER BY
21     latest_treatment.visit_date;
22

```

	pet_name character varying (100)	treatment_type character varying (300)	visit_date timestamp without time zone
1	Napoleon	Podanie tabletki	2019-04-20 09:00:00
2	Klatka	Nawodnienie	2019-07-11 12:00:00
3	Dakota	Zabieg	2020-12-12 17:00:00
4	Lili	Operacja	2021-07-20 09:00:00

Zapytanie 2

Zapytanie drugie dotyczyło ostatnich wizyt weterynarzy pracujących w klinice. Wraz z datą ostatnich wizyty, zwrócono także imię zwierzęcia które zostało przyjęte podczas tej wizyty oraz liczbę wykonanych podczas niej procedur medycznych i postawionych diagnoz.

```

1  -- Query 2
2
3
4  SELECT
5      staff.first_name AS vet_first_name,
6      staff.last_name AS vet_last_name,
7      visit.visit_date,
8      pet.pet_name,
9      COALESCE(treatment_count.treatments_done, 0) AS treatments_done,
10     COALESCE(diagnosis_count.diagnoses_given, 0) AS diagnoses_given
11 FROM
12     visit
13 JOIN pet ON visit.pet_id = pet.pet_id
14 JOIN staff ON visit.vet_id = staff.staff_id
15 LEFT JOIN (
16     SELECT treatment.visit_id, COUNT(*) as treatments_done
17     FROM treatment
18     GROUP BY treatment.visit_id
19 ) AS treatment_count ON visit.visit_id = treatment_count.visit_id
20 LEFT JOIN (
21     SELECT diagnosis.visit_id, COUNT(*) as diagnoses_given
22     FROM diagnosis
23     GROUP BY diagnosis.visit_id
24 ) AS diagnosis_count ON visit.visit_id = diagnosis_count.visit_id
25 WHERE
26     visit.visit_id IN (
27         SELECT v2.visit_id
28         FROM visit v2
29         WHERE v2.vet_id = visit.vet_id
30         AND v2.visit_date = (
31             SELECT MAX(v3.visit_date)
32             FROM visit v3
33             WHERE v3.vet_id = v2.vet_id
34             GROUP BY v2.vet_id
35         ))
36 ORDER BY
37     staff.last_name, staff.first_name, visit.visit_date;
38

```

-- Query 2

```

SELECT
    staff.first_name AS vet_first_name,
    staff.last_name AS vet_last_name,
    visit.visit_date,
    pet.pet_name,
    COALESCE(treatment_count.treatments_done, 0) AS treatments_done,
    COALESCE(diagnosis_count.diagnoses_given, 0) AS diagnoses_given
FROM
    visit

```

Data Output Messages Notifications

	vet_first_name character varying (100)	vet_last_name character varying (100)	visit_date timestamp without time zone	pet_name character varying (100)	treatments_done bigint	diagnoses_given bigint
1	Artur	Janowicz	2022-11-13 13:00:00	Klatka	0	1
2	Anita	Mazurczak	2021-07-22 11:00:00	Laweta	0	0
3	Juliana	Mrotek	2020-07-23 14:00:00	Lea	0	0
4	Jakub	Sierakowski	2023-04-16 12:00:00	Dakota	0	0
5	Sylvia	Witowska	2021-07-20 09:00:00	Lili	1	0

Indeksy

Stworzono następujące indeksy:

Dla tabeli diagnosis, stworzone zostały: indeks na kolumnie 'visit_id', indeks na kolumnie 'diagnosis_id'. Indeksy te wynikają z potrzeby częstego wykonania operacji join z tabelą visit w celu sprawdzenia kiedy dana diagnoza została postawiona oraz operacji join z medicine_plan, aby sprawdzić jakie plany lekowe zostały wprowadzone w ramach leczenia zdiagnozowanej choroby. Szukanie diagnoz po ich typie czy uwagach do nich dołączonych nie ma sensu ze względu na dużą różnorodność ich wartości.

Dla tabeli medicine_plan: indeks na kolumnie diagnosis_id. Tak jak wspomniano, indeks wynika z częstej potrzeby wykonania operacji join z tabelą diagnosis_id. Indeksy na inne kolumny nie są potrzebne, ponieważ dany plan lekowy interesuje weterynarzy i właścicieli zwierząt głównie z perspektywy adresowanej diagnozy.

Dla tabeli staff_position: indeks na kolumnie staff_id, indeks na kolumnie position_id. Dla tabeli staff: indeks na kolumnie staff_id. Dla tabeli position: indeks na kolumnie position_id. Indeksy te stworzone zostały ze względu na częste operacje join między tabelami staff_position i position oraz między tabelami Staff_position i Staff. Wynika to z ścisłego powiązania reprezentacji tych tabel (position reprezentuje stanowisko, staff_position reprezentuje zatrudnienie pracownika na danym stanowisku a staff, samego pracownika).

Dla tabeli pet_measurement: indeks kompozytowy na kolumnach pet_id, measurement_date. Wynika to z tego, że najczęściej obchodzą nas ostatnie pomiary zwierzęcia lub pomiary w danym czasie. Również w widokach dla weterynarzy i właścicieli na tabelę zwierząt wyświetlone są ostatnie pomiary wagi i wzrostu zwierzęcia.

Dla tabeli pet: indeks na kolumnie pet_id, indeks na kolumnie owner_id. Dla tabeli pet_owner: indeks na kolumnie owner_id. Wybór indeksów dla tabel pet i pet_owner wynika z potrzeby częstego przypisania właściciela zwierzęcia do zwierzęcia, również w widokach przygotowanych dla użytkowników.

Dla tabeli visit: indeks kompozytowy na kolumnach pet_id i visit_date oraz indeks na kolumnie visit_id. Indeks kompozytowy wynika z częstej potrzeby znalezienia wizyty danego zwierzęcia w określonym terminie lub znalezienia ostatniej wizyty. Indeks na kolumnie visit_id wynika z częstej operacji join np. z tabelą pet lub treatment. Indeksy kompozytowe pozwalają na przyspieszenie wyszukiwania w zapytaniach w których odnosimy się do wartości więcej niż jednej kolumny, np. poprzez użycie AND.

Testy:

W celu przetestowania działania indeksów, dodano więcej danych syntetycznych i uruchomiono zapytanie 1 i 2 opisane powyżej.

Dla zapytania 1:

Indeksy	Średnia czasu wykonania zapytania (dla 5 zapytań)
Bez indeksów	85.4 msec
Z wszystkimi opisanymi powyżej indeksami	81.2 msec
Jedynie z indeksem kompozytowym na tabeli visit	83 msec
Jedynie z indeksem visit_id na tabeli visit	85.2 msec

Dla zapytania 2:

Indeksy	Średnia czasu wykonania zapytania (dla 5 zapytań)
Bez indeksów	85 msec
Z wszystkimi opisanymi powyżej indeksami	77.2 msec
Jedynie z indeksem kompozytowym na tabeli visit	79.8 msec
Jedynie z indeksem visit_id na tabeli visit	84 msec