

## ETAP II

### **2. Analiza liczby instancji – dla stowarzyszenia „START”**

Poniższa analiza liczby instancji została początkowo przeprowadzona na potrzeby stowarzyszenia „START”, dla którego aplikacja jest opracowywana. W kolejnym rozdziale przeanalizujemy bardziej ogólną postać bazy danych.

Zakładamy, że projektujemy system, który działać będzie przez 10 lat.

Tabela 1 Analiza liczby instancji

<b>Nazwa encji</b>	<b>Liczba instancji</b>
Zawodnicy	250
Sekcje	10
Trenerzy	25
Treningi	6000
Zawody	300
Składki	250
Orzeczenia	250
Płatności	5000
Komunikaty	600
Sprawozdania	6300
Trofea	900

Obliczenia:

Zawodnicy – 250

Sekcje – 10

Trenerzy – 25 (dla każdej sekcji dwóch trenerów + 5 zapasowych)

Treningi – 6000 dla (na 10 lat, dla każdej sekcji 1 w tygodniu + zapas)

Zawody – 300 (dla każdej sekcji 2x w roku, 10 sekcji \*2 \*10 lat + 100 dodatkowych)

Składki – 250 (tyle co zawodników)

Orzeczenia – 250 (tyle co zawodników)

Płatności – 5000 (250 zawodników \* 2 możliwe raty \* 10lat)

Komunikaty – 600

W ciągu 1 roku:

30 komunikatów odnośnie zawodów

500 komunikatów o zapłacie składki

250 komunikatów o wygasaniu orzeczenia

6000 komunikatów odnośnie treningów

7000 (z zapasem)

W ciągu 1 miesiąca (30 dni ważności komunikatu) ok. 600 komunikatów.

Sprawozdania – 6300 (tyle co zawodów i treningów)

Trofea – 900 (trzy trofea [medale] dla każdego zawodów)

### **3. Analiza użycia i integralności – stowarzyszenie „START”**

Na podstawie uprawnień, jakie dani użytkownicy (administrator, księgowy czy trener) będą posiadać, przygotowaliśmy zestawienie pokazujące, w jakim stopniu będą używane wybrane tabele (encje) w bazie danych. Następnie, zgodnie z przyjętym przez nas stopniem obciążenia, które generuje dana operacja, obliczyliśmy wartości mówiące, na których tabelach będzie wykonywana największa liczba operacji (zostało to przedstawione na wykresie).

*Tabela 2 uprawnienia do danych tabel nadane odpowiednim użytkownikom*

	Wstawianie (admin księgowy trener)			Usuwanie (admin księgowy trener)			Modyfikacja (admin księgowy trener)			Wyszukiwanie (admin księgowy trener)		
	+	-	-	+	-	-	+	-	+	+	+	+
<b>Zawodnicy</b>	+	-	-	+	-	-	+	-	+	+	+	+
<b>Sekcje</b>	+	-	-	+	-	-	+	-	-	+	+	+
<b>Trenerzy</b>	+	-	-	+	-	-	+	-	-	+	+	-
<b>Treningi</b>	+	-	+	+	-	-	+	-	+	+	+	+
<b>Zawody</b>	+	-	+	+	-	-	+	-	+	+	+	+
<b>Składki</b>	+	+	-	+	+	-	+	+	-	+	+	-
<b>Orzeczenia</b>	+	-	-	+	-	-	+	-	-	+	+	-
<b>Płatności</b>	+	+	-	+	+	-	+	+	-	+	+	-
<b>Komunikaty</b>	+	+	+	+	-	-	+	+	+	+	+	+
<b>Sprawozdania</b>	+	-	+	+	-	-	+	-	+	+	+	+
<b>Trofea</b>	+	-	+	+	-	-	+	-	-	+	+	+



Rysunek 1 Wykres zmienności (używalności) poszczególnych tabel

Projektując bazę danych dla stowarzyszenia „START”, które jest przedsiębiorstwem stosunkowo małym i nie będzie generować dużych obciążeń do bazy danych, a jego praca będzie raczej statyczna, analiza integralności jest w zasadzie zbędna. Sprowadzi się ona do zadbania o odpowiednią kontrolę typów danych w tabelach bazy oraz o nadawanie użytkownikom prawidłowych uprawnień, co pokazaliśmy w Tabeli 2.

Podczas użytkowania aplikacji przez stowarzyszenie nie wystąpią także żadne anomalie czy sytuacje szczególne, np. wykonywane jedynie o wyznaczonej porze dnia.

#### **4. Analiza użycia i integralności – przypadek ogólny**

Na wykresie ukazującym zmienności zawartości poszczególnych tabel widoczne jest, które tabele generować będą największe obciążenie dla bazy danych. Gdybyśmy założyli, że nasza aplikacja projektowana jest do celów ogólnych, tzn. nie tylko dla wybranego przez nas przedsiębiorstwa, kiedy musiałaby zostać ona dostosowana na etapie projektowym do potrzeb dowolnej firmy/stowarzyszenia o zbliżonym do „STARTu” profilu działalności, mogłyby pojawić się pewne problemy z integralnością i prawidłowym działaniem takiej bazy.

Jeśli założymy, że liczba instancji w bazie danych wzrośnie kilka lub kilkanaście razy, mogą pojawić się następujące problemy:

- **Komunikaty** – jeśli liczba wysyłanych każdego dnia komunikatów drastycznie wzrośnie, obciążenie zadawane bazie danych może okazać się zbyt duże; system monitorujący, który o wyznaczonych porach dnia będzie przeglądał bazę komunikatów w celu wysłania niewysłanych jeszcze wiadomości może mieć z tym problem, gdyż będzie musiał wysłać bardzo wiele komunikatów, wykonując jednocześnie zapytania do wielu innych tabel, co może być powodem problemów z poprawnym działaniem całej bazy danych w tym czasie; rozwiązanie: sortowanie komunikatów wg wysłanych/niewysłanych, np. niewysłane jako pierwsze w tabeli i wprowadzenie indeksu, który wskazywałby na pierwszy/ostatni niewysłany element
- **Logowanie do systemu** – jeśli baza danych będzie miała dużo więcej użytkowników z różnymi poziomami uprawnień, pojawi się problem logowania oraz otwierania sesji dla aplikacji bazodanowej; w przypadku rozbudowanego, dużego przedsiębiorstwa, które posiadało będzie różne działy, a więc także różne grupy pracowników, może wystąpić sytuacja, gdy o danych porach dnia grupy te będą wykonywały logowanie do systemu; momenty te generować będą określone obciążenie dla bazy danych, jednak występować będą jedynie o określonych porach dnia
- **Zapytania** - w przypadku rozrostu bazy naturalnie wykonywana będzie większa liczba zapytań do tabel z danymi; może to być powodem do zastosowania innych rozwiązań, technologii lub poziomów uprawnień dla użytkowników, które ograniczyłyby niepotrzebne zapytań; rozwiązanie: dobre zindeksowanie tablic w celu przyspieszenia dostępu do danych

## **5. Wybór SZBD**

System Zarządzania Bazą Danych, który wybraliśmy do fazy implementacyjnej naszego projektu to system **Oracle MySQL**. Mieliśmy do czynienia z tym systemem w IV semestrze, dlatego wydaje się to naturalnym wyborem.

SZBD firmy Oracle zapewnia wygodny, dostępny przez przeglądarkę internetową interfejs, który oprócz utworzenia struktury samej bazy poprzez zestaw zapytań pozwala na stworzenie webowej aplikacji obsługującej utworzoną bazę, co pozwoli na pewne przygotowanie przed końcowym etapem, jakim będzie utworzenie desktopowej aplikacji bazodanowej.

## **ETAP III**

### **6. Fizyczny projekt bazy danych**

Baza wraz ze wszystkimi tabelami, sekwencjami, triggerami oraz pozostałymi niezbędnymi elementami została zaimplementowana w wybranym przez nas środowisku Oracle. Baza działa na lokalnym serwerze Oracle. Diagram związków pomiędzy gotowymi tabelami dołączony jest do opisu niniejszego etapu projektu. Załączony został również skrypt SQL generujący całą stworzoną bazę danych. W modelu fizycznym warto zwrócić uwagę na dwie tabele utworzone w celu zapisywania informacji o logach systemowych (LOGI\_PLATNOSCI, LOGI\_ZAWODNICY). Dołączony został także plik z zapytaniem generującym losowe dane do testów.

### **7. Zbiór zapytań zoptymalizowanych, indeksowanie, testy wydajnościowe**

W celu podniesienia wydajności projektowanej przez nas bazy danych oraz umożliwienia dostatecznie szybkiego dostępu do danych w tabelach, przeanalizowaliśmy stopień użycia naszych struktur i doszliśmy do wniosku, że konieczne jest utworzenie następujących indeksów:

- a) indeks w tabeli KOMUNIKATY na kolumnie CZY\_WYSLANY – pozwoli to systemowi wysyłającemu niewysłane jeszcze do tej pory komunikaty na szybsze odnalezienie pierwszego oraz pozostałych niewysłanych komunikatów
- b) indeks w tabeli ORZECZENIA na kolumnie DATA\_WAZNOSCI – kolumna ta będzie często używana w klauzuli WHERE, czy to przez system monitorujący w celu przypominania o kończącym się orzeczeniu, czy też generowania raportów związanych z ważnością orzeczeń.

Intensywność eksploatacji bazy danych nie jest na tyle złożona, aby jej obsługa wymagała dodatkowego indeksowania.

W celu przeprowadzenia testów wydajnościowych bazy oraz jej reakcji na złożone zapytania wszystkie tabele w bazie danych zostały wypełnione wieloma wartościami losowymi:

*Tabela 3 Liczba instancji testowych dla poszczególnych tabel*

<b>Tabela</b>	<b>Liczba instancji testowych</b>
Orzeczenia	200000
Płatności	150000
Sekcje	100
Składki	150000
Sprawozdania	300
Trenerzy	60
Trofea	2500
Zawodnicy	100000
Zawodnicy_Orzeczenia	200000
Zawodnicy_Treningi	50000
Zawody	300
Zawody_Zawodnicy	1000000

Wszystkie zapytania i pomiary czasów ich trwania wykonywane były na bazie Oracle działającej na serwerze lokalnym (*localhost*), na komputerze przenośnym Lenovo Y580 wyposażonym w procesor Intel Core i5-3210M 2x 2.50GHz oraz 8 GB pamięci RAM, pracującego pod kontrolą systemu operacyjnego Windows 7 Professional x64.

Przykładowe czasy wypełniania tabel danymi losowymi:

- Orzeczenia (200000 instancji): 6,21 sekundy
- Trenerzy (60 instancji): 0,03 sekundy
- Zawodnicy (100000 instancji): 22,21 sekundy
- Płatności (150000 instancji): 27,25 sekundy
- Składki (150000 instancji): 8,14 sekundy.

Czasy dodawania rekordów są zatem niezauważalne dla użytkownika. Sprzęt, na którym dokonano testów posiada jeszcze spory „zapas” mocy jak na obsługę tych wymiarów bazy danych.

Po wypełnieniu wszystkich tabel bardzo dużymi (jak na zastosowania naszej bazy danych) liczbami wierszy wykonane zostały testy wydajnościowe poprzez wykonywanie na bazie złożonych poleceń, które wymagały łączenia 2, 3 a nawet 4 tabel w jednym zapytaniu:

- a) Wyświetlenie ID oraz nazwy sekcji, a także liczby zawodników do danej sekcji przypisanych (łączenie tabel SEKCJE oraz ZAWODNICY):

```
SQL >> SELECT s.id_sekcji "ID Sekcji", s.nazwa "nazwa sekcji",
z."liczba zawodnikow" FROM sekcje s JOIN (SELECT sekcja, COUNT(*)
"liczba zawodnikow" FROM zawodnicy GROUP BY sekcja) z ON
s.id_sekcji = z.sekcja;
```

**Wynik: 99 wierszy Czas wykonania: 0,21 sekundy**

- b) Wyświetlenie nazwiska zawodnika, ID każdego treningu, w którym brał udział oraz nazwy sekcji, do której jest przypisany (łączenie tabel ZAWODNICY, SEKCJE oraz TRENINGI):

```
SQL >> SELECT z.nazwisko "nazwisko zawodnika", t.id_treningu "ID
Treningu", s.nazwa "Nazwa sekcji" FROM Zawodnicy z JOIN
Zawodnicy_treningi t ON z.id_zawodnika=t.id_zawodnika JOIN Sekcje
s ON z.sekcja=s.id_sekcji;
```

**Wynik: 50000 wierszy Czas wykonania: 2,00 sekundy**

- c) Jak w punkcie b), ale dodatkowo wyświetlenie daty ważności orzeczenia dla każdego wymienionego zawodnika (łączenie tabel ZAWODNICY, SEKCJE, TRENINGI oraz ORZECZENIA):

```
SQL >> SELECT z.nazwisko "nazwisko zawodnika", t.id_treningu "ID
Treningu", s.nazwa "Nazwa sekcji", o.data_waznosci "Data waznosci
orzeczenia" FROM Zawodnicy z JOIN Zawodnicy_treningi t ON
z.id_zawodnika=t.id_zawodnika JOIN Sekcje s ON
z.sekcja=s.id_sekcji JOIN Orzeczenia o ON z.orzeczenie =
o.id_orzeczenia;
```

**Wynik: 50000 wierszy Czas wykonania: 2,48 sekundy**

Czasy trwania generowania złożonych zapytań wraz z otrzymaniem odpowiedzi pokazały, że baza danych jest wydajna oraz dobrze zoptymalizowana. Pozwoliło nam to na odrzucenie rozważań na temat wprowadzania dalszej indeksacji czy głębszej normalizacji bazy danych (jak na przykład utworzenie osobnej tabeli ADRESY, która w naszych zastosowaniach okazała się niepotrzebna, ponieważ adres nie będzie często wykorzystywany, nie będą również generowane żadne statystyki/raporty na podstawie składowych adresu).

## 8. Polityka bezpieczeństwa

W ramach analizy bezpieczeństwa naszej bazy danych, a także związanej z nią później aplikacji oferującej dostęp do jej zasobów przeanalizowaliśmy, co w tej kwestii może okazać się istotne dla naszego systemu:

- a) należy dbać o **aktualność oprogramowania systemu bazy danych**, weryfikować, czy instalowane aktualizacje pochodzą z zaufanego źródła (od producenta), gdyż zainstalowanie aktualizacji z obcego źródła może umożliwić przejęcie kontroli nad bazą danych; oprogramowanie najlepiej aktualizować manualnie, nie korzystać ze zautomatyzowanych skryptów instalacyjnych; bazę działającą na nowej wersji oprogramowania sprawdzić wcześniej na serwerze testowym
- b) ochrona **na poziomie samej bazy danych**:
  - świadomość, że najcenniejszymi i podlegającymi najściślejszej ochronie danymi są przechowywane dane osobowe
  - nadawanie **odpowiednich uprawnień** użytkownikom systemu: użytkownik systemu, z użyciem którego uruchomiona jest baza danych nie powinien posiadać uprawnień administratora (root) w systemie operacyjnym; nie należy nadawać użytkownikom niepotrzebnych lub wykraczających poza ich kompetencje uprawnień [WYKONANO W NASZEJ BAZIE DANYCH]; unikać oferowania połączeń typu SSH na konto użytkownika bazy danych
  - administrator powinien mieć dostęp do bazy **tylko z lokalnej podsieci**
  - sprawdzenie, czy dostęp nie jest zapewniony za pomocą kont tworzonych z hasłami domyślnymi (w systemie Oracle np. kombinacja DBSNMP/DBSNMP)
  - upewnienie się, że na serwerze nie zostały zainstalowane domyślne bazy typu „demo”
  - **logi systemowe**: rejestrowanie prób logowania (udane, nieudane), operacji na BD wraz z datami; składować logi w tabelach o ograniczonym dostępie dla użytkowników i wykonywać ich kopie zapasowe [WYKONANO W NASZEJ BAZIE DANYCH]
  - szyfrowanie przechowywanych danych (szczególnie danych osobowych); rozwiązanie to sprawdza się szczególnie w niewielkich bazach danych; w dużych magazynach danych należy rozważyć, czy zaszyfrowanie nie spowoduje niedopuszczalnych opóźnień w działaniu systemu
- c) ochrona **na poziomie sprzętowym**:
  - jeśli bazę danych trzymamy na lokalnym serwerze lub komputerze, **warto wydzielić osobną od systemowej partycję na dysku do przechowywania zasobów BD**; w przeciwnym wypadku, gdyby baza przechowywana była na jednej partycji z systemem operacyjnym, wypełnienie partycji systemowej mogłoby spowodować niestabilność lub brak odpowiedzi systemu
  - upewnienie się, że używamy prawidłowego systemu plików, np. NTFS zamiast FAT (NTFS umożliwia nadawanie uprawnień do odczytu/zapisu)



- rozważenie wprowadzenia technologii typu RAID w przypadku awarii dysku twardego
- d) ochrona **na poziomie aplikacji dostępowej**:
- logowanie: **zabezpieczenie przed atakami typu SQL Injection** poprzez kontrolę dozwolonych do wprowadzania znaków w polach logowania (ograniczenie możliwości używania nawiasów oraz słów kluczowych SQL)
  - **szyfrowanie** przesyłanych do i z bazy danych danych logowania
  - zwrócenie uwagi, czy na stacji roboczej, z której następuje logowanie do bazy danych za pośrednictwem aplikacji klienta nie występuje przechowywanie danych logowania, np. w Rejestrze Windows i czy dane te nie są danymi użytkownika mającego pełny dostęp do BD.

W zależności od zastosowań, sposobu przechowywania oraz rozmiarów systemu bazy danych z wymienionych powyżej aspektów bezpieczeństwa i mechanizmów obronnych wybiera się niektóre i wprowadza do systemu. W naszej bazie danych zdecydowaliśmy się na wprowadzenie następujących zabezpieczeń:

- utworzenie kont użytkowników wraz z odpowiednimi dla nich uprawnieniami zgodnie z utworzoną wcześniej Tabelą 2.
- zapisywanie logów:
  - a) tabela LOGI\_PLATNOSCI - przechowuje logi zmiany w tabeli PLATNOSCI wraz z datą oraz rodzajem wykonanej zmiany (dotyczy zmiany kwoty płatności lub przypisanego do niej zawodnika)
  - b) tabela LOGI\_ZAWODNICY – przechowuje informacje o wszelkich zmianach, wstawieniach lub usunięciach z tabeli ZAWODNICY wraz z datą dokonania zmiany oraz jej rodzajem.

Do tworzenia logów wykorzystywane są odpowiednio skonfigurowane triggery.

Dodatkowo wykonaliśmy prostą aplikację w Javie, która korzystając ze sterownika JDBC łączy się z bazą danych (jest możliwe logowanie jako dowolny użytkownik) oraz wysłanie dowolnego zapytania do bazy danych. Prawdopodobnie w tej technologii będziemy pracować podczas ostatniego etapu projektu.