

Comparative analysis of database systems dedicated for Android

Analiza porównawcza systemów baz danych dedykowanych dla systemu Android

Kamil Wałachowski*, Grzegorz Kozieł

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The article presents a comparative analysis of mobile databases dedicated for Android. The comparative analysis was carried out on the example of a relational SQLite database with selected nonrelational databases: Realm, ObjectBox and SnappyDB. Theoretical issues were discussed, such as stored data types. Research was carried out to check the performance of mobile databases in terms of: saving, editing, deleting, searching and sorting data. CPU and RAM usage were examined during saving data. The research also included a comparison the size of the database files on the internal disk.

Keywords: ObjectBox; Realm; SQLite; SnappyDB

Streszczenie

W artykule przedstawiona została analiza porównawcza mobilnych baz danych dedykowanych dla systemu Android. Analizę porównawczą przeprowadzono na przykładzie relacyjnej bazy danych SQLite z wybranymi nierelacyjnymi bazami danych: Realm, ObjectBox oraz SnappyDB. Omówione zostały zagadnienia teoretyczne takie jak przechowywane typy danych. Przeprowadzone zostały badania, mające na celu sprawdzenie wydajności mobilnych baz pod kątem: zapisu, edycji, usuwania, wyszukiwania oraz sortowania danych. Zbadane zostało wykorzystanie procesora oraz pamięci RAM podczas zapisu danych. Badania objęły także porównanie wielkości plików baz danych znajdujących się na dysku wewnętrznym.

Słowa kluczowe: ObjectBox; Realm; SQLite; SnappyDB

*Corresponding author

Email address: kamil.walachowski@gmail.com (K. Wałachowski)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Co roku obserwowany jest znaczący wzrost liczby urządzeń mobilnych znajdujących się w posiadaniu ludzi. Liczba aktywnych kart sim wydanych przez polskich operatorów telekomunikacyjnych jest większa od liczby mieszkańców Polski [1,2]. Świadczy to o dużym zapotrzebowaniu na urządzenia mobilne i dostępne za ich pomocą usługi. Ponad 92% Polaków korzysta z telefonów komórkowych [3]. Ich podstawowym zastosowaniem jest komunikacja [3]. Jednak możliwości urządzeń mobilnych, zbliżone do komputerów stacjonarnych, pozwalają na użycie ich na wiele innych sposobów, o czym świadczy bogactwo różnorodnych aplikacji dostępnych na tego typu urządzenia [4].

Wiele z tych aplikacji wymaga przechowywania danych na urządzeniu mobilnym. Realizacja tego zadania możliwa jest na wiele sposobów. Jednak istotne jest zapewnienie możliwości trwałego zapisu danych, oferującego możliwość łatwego i wydajnego dostępu do nich, wyszukiwania oraz modyfikowania treści, a także ich usuwania w razie potrzeby. Funkcje te oferują bazy danych, które stały się nieodzowne w każdej niemal aplikacji.

Początkowo wszystkie aplikacje mobilne korzystały z jedynej dostępnej na rynku relacyjnej bazy danych SQLite. Obecnie na rynku pojawiają się nowe bazy NoSql, mogące w przyszłości zastąpić popularną bazę SQLite. Bazy NoSql udostępniają mechanizm przechowywania danych w bazach, w których sposób modelowania jest inny niż tabelaryczny. W niniejszym artykule podjęto się analizy porównawczej czterech wybranych baz danych dostępnych dla urządzeń pracujących pod kontrolą systemu operacyjnego Android [5].

2. Mobilne bazy danych

SQLite to biblioteka implementująca niezależny, bezserwerowy, bezkonfiguracyjny silnik bazy danych. Kod SQLite jest własnością publiczną, a zatem jest darmowy do użytku w dowolnym celu, komercyjnym lub prywatnym. SQLite to wbudowany silnik bazy danych SQL (ang. Structured Query Language). W przeciwieństwie do większości innych baz danych SQL, SQLite nie ma osobnego procesu serwera. SQLite czyta i zapisuje dane bezpośrednio do zwykłych plików na dysku. Kompletna baza danych SQL z wieloma tabelami, indeksami, wyzwalaczami i widokami znajduje się w jednym pliku na dysku. Format pliku bazy danych jest wieloplatformowy, swobodnie można przenosić bazę danych między systemami 32-bitowymi i 64-bitowymi [6].

Realm jest to obiektowa baza danych NoSql zaprojektowana na platformę mobilną. Jej rdzeń składa się z niezależnej biblioteki C++. Główną zaletą bazy Realm jest prostota, dane są bezpośrednio ujawniane jako obiekty i mogą być wyszukiwane za pomocą kodu. Realm do przechowywania plików wykorzystuje własne pliki, które można bezpiecznie przenosić między syste-

mami Android oraz iOS. Realm umożliwia łatwą pracę z wieloma wątkami, umożliwia także szyfrowanie danych przy pomocy AES-256 [7].

ObjectBox jest to lokalna baza danych, zorientowana obiektowo, wspierająca obsługę relacji. Baza ta obsługuje wiele platform i języków opartych na JVM (ang. Java Virtual Machine), takich jak Java oraz Kotlin. ObjectBox, podobnie jak Realm, odznacza się prostotą pisania kodu, posiada wiele gotowych funkcji ułatwiających pracę. Ponadto ObjectBox wspiera wielowątkowość. Wszystkie dane zapisywane są w pojedynczym pliku bazy danych [8].

SnappyDB jest najmniej zaawansowaną bazą danych spośród prezentowanych. SnappyDB jest to darmowa baza typu klucz-wartość bazująca na bazie LevelDB, dedykowana dla systemu Android. Pozwala ona przechowywać i pobierać prymitywne typy, pozwala także na serializację obiektów i tabel w bezpieczny sposób. SnappyDB nie wspiera transakcji, relacji, wielowątkowości. Cała baza mieści się w jednym pliku znajdującym się w pamięci wewnętrznej urządzenia mobilnego [9].

Wersje testowanych baz:

- Realm 5.12.0;
- SnappyDB 0.4.0;
- SQLite 3.8;
- ObjectBox 2.3.4.

3. Przegląd literatury

Trevor Perrier oraz Fahad Pervaiz w artykule „SQL in a Mobile World Benchmarking Embedded Mobile Databases” dokonują oceny trzech typów modeli przechowywania danych na urządzeniach mobilnych. Testy zostały przeprowadzone na bazie relacyjnej SQLite, bazie typu klucz-wartość CouchLite oraz bazie obiektowej DbO4. Autorzy opracowali kompleksowy zestaw testów, aby przetestować każdą bazę danych w aplikacji na systemie Android. Testy porównawcze polegały na utworzeniu plików bazy danych, skonfigurowaniu dowolnego niezbędnego schematu, a następnie uruchomieniu każdego z sześciu zapytań typu CRUD dziesięć razy. Przeprowadzone przez autorów testy wyraźnie pokazują, że model relacyjny ma najlepsze wyniki przy dużym obciążeniu i najbardziej skomplikowanych zapytaniach. Baza obiektowa DbO4 była najszybsza w przypadku tworzenia, wstawiania oraz odczytu prostych zapytań [10].

W artykule „Mobile Data Store Platforms: Test Case based Performance Evaluation” K. Kussainova i B. Kumalakova dokonano porównania baz typu SQL oraz NoSql. Na potrzeby przeprowadzanych testów autorzy wybrali bazę relacyjną SQLite, bazę obiektową Realm oraz bazę typu klucz-wartość SnappyDB. Bazy zostały zbadane pod kątem zużycia pamięci, czasu wykonywania funkcji zapisu oraz odczytu. Zostały stworzone zapytania odczytujące nawet 12000 rekordów. Według przeprowadzonych testów Realm okazał się bazą najszybciej odczytującą proste zapytania, aż 12000 rekordów w 4ms. Badania wykazały także, że

SnappyDB ma najmniejsze zużycie pamięci RAM, średnio 12MB [11].

W artykule „A Comparison of Lightweight Databases in Mobile Systems” twórcy badają systemy baz danych specjalnie zaprojektowane i zoptymalizowane dla systemów mobilnych. Autorzy pracy zwracają uwagę na ograniczenia systemów mobilnych - pamięć oraz źródło energii (baterie). Praca opisuje konieczność zastosowania tzw. lekkich systemów bazodanowych oraz przedstawia i krótko charakteryzuje każdą z trzech baz danych - SQLite, DbO4 oraz H2. Do porównania wcześniej wymienionych baz wykorzystane zostało narzędzie PolePosition13. Narzędzie to posiada pięć standardowych testów porównawczych do oceny szybkości działania baz danych. Wykonano trzynaście różnych testów mających na celu sprawdzenie wydajności baz. Na podstawie przeprowadzonych badań autorzy stwierdzili, że baza relacyjna H2 jest najszybszą i najbezpieczniejszą bazą danych wykorzystywaną na urządzeniach mobilnych [12].

Artykuł „Porównanie możliwości wykorzystania oraz analiza wydajności baz danych na systemach mobilnych” przedstawia formy składowania danych na urządzeniach mobilnych oraz opisuje sposoby połączeń z zewnętrznymi systemami bazodanowymi takimi jak MySQL, PostgreSQL a także Microsoft SQL Server. Twórcy w swojej pracy starają się odpowiedzieć na pytanie, który sposób przechowywania danych ma największy sens w poszczególnych przypadkach. Do przeprowadzenia testów wykorzystane zostały dwa urządzenia mobilne: jeden z systemem Android, drugi z systemem Windows Phone. Badanie zapisu oraz odczytu zostało przeprowadzone dla:

- par klucz-wartość;
- rekordów z danymi tekstowymi;
- rekordów z danymi numerycznymi;
- rekordów z danymi binarnymi.

Zapisywanych oraz odczytywanych było kolejno 10, 100, 1000, 10000, 100000 oraz 1000000 rekordów. Na podstawie przeprowadzonych badań na obu telefonach, autorzy zauważyli, że najdłuższe czasy wykonywania operacji zapisu oraz odczytu notuje lokalna baza SQLite. Autorzy spostrzegli także, że wszystkie testowane bazy danych są efektywne do ok. 10000 rekordów, wówczas czasy wykonywania operacji odczytu i zapisu nie przekraczają kilkunastu sekund. Badania wykazały, że najlepszą bazą znajdującą się na zewnętrznym serwerze jest Microsoft SQL Server [13].

4. Metodyka badawcza

4.1. Scenariusze badawcze

Badaniom poddane zostały wybrane bazy danych dostępne dla systemu Android. Badania zostały przeprowadzone na czterech bazach: Realm, SQLite, ObjectBox oraz SnappyDB. Celem przeprowadzenia badań stworzony został zestaw danych. Dane zostały wygenerowane automatycznie przy pomocy darmowej biblioteki Java Faker [14]. Wszystkie niżej opisane badania zostały przeprowadzone pięciokrotnie na dwóch telefo-

nach, w celu otrzymania rzetelnych wyników. Testy przeprowadzane były na urządzeniach znajdujących się w stanie optymalnym. Oznacza to, że wszystkie aplikacje działające w tle zostały wyłączone, działały tylko niezbędne usługi. Dodatkowo przed wykonaniem każdej próby, wyczyszczone zostały dane aplikacji oraz pamięć podręczna.

Operacja zapisu danych została przeprowadzona z wykorzystaniem transakcji w technologiach, które to umożliwiają. Zbadany został czas dodawania kolejno 100, 1000, 10000 oraz 100000 rekordów. SnappyDB nie wspiera transakcji, dlatego dodawanie elementów odbywało się pojedynczo przy pomocy pętli for.

Operacje usuwania oraz edycji danych zostały wykonane na bazie zawierającej 100000 rekordów. Zmierzony został czas usuwania, a także edycji 100, 1000, 10000 oraz 100000 rekordów z wykorzystaniem transakcji. W przypadku operacji edycji aktualizowane było pole imię. Analogicznie jak w przypadku operacji dodawania, usuwanie i aktualizacje danych w bazie SnappyDB odbywała się przy pomocy pętli for.

Operacje wyszukiwania danych zostały przeprowadzone na bazie zawierającej 100000 rekordów. Pierwszy test obejmował wyszukiwanie danych po ciągu znaku, z tabeli wyszukane zostało słowo kluczowe 'monitor'. Kolejne badanie miało za zadanie sprawdzenie prędkości odczytu zapytań warunkowych. Eksperymenty wyszukiwania nie zostały wykonane na bazie SnappyDB, ponieważ nie wspiera on wykonywania zapytań złożonych. Poniżej przedstawione zostały zapytania wykorzystane do eksperymentu:

1. `SELECT * FROM users WHERE hardware = "monitor";`
2. `SELECT * FROM users WHERE age > 10 AND CONTAINS(country, "Ja") LIMIT 10;`
3. `SELECT DISTINCT country FROM users WHERE weight > 50 AND height < 180;`

Operacja sortowania wykonana została na bazie zawierającej 100000 rekordów. Badanie to zostało wykonane dla baz umożliwiających sortowanie – SQLite, Realm oraz ObjectBox. Posortowany został wiek użytkowników od najmniejszego do największego.

Ostatnie badanie miało na celu sprawdzenie wykorzystania zasobów urządzenia mobilnego. Testy obejmowały sprawdzenie wykorzystania pamięci RAM oraz procesora podczas zapisu 100000 rekordów do pustej bazy. Dodatkowo sprawdzone zostały rozmiary plików baz danych znajdujących się na dysku wewnętrznym telefonu.

4.2. Stanowisko badawcze

Badania przeprowadzone zostały na dwóch urządzeniach mobilnych o różnej specyfikacji. Pierwszym z nich był smartfon HUAWEI P9 Lite mini o specyfikacji:

- CPU: Qualcomm Snapdragon 425 8917, 4x1,4 GHz;
- Pamięć RAM: 2 GB;
- GPU: Adreno 308 @500 MHz;
- Pamięć wbudowana: 16GB;
- System: Android 7.0 Nougat.

Drugim urządzeniem, na którym przeprowadzono badania, był HUAWEI P10 Lite mini o specyfikacji:

- CPU: HiSilicon Kirin 658, 8x2,1 GHz;
- Pamięć RAM: 4 GB;
- GPU: ARM Mali-T830 MP2;
- Pamięć wbudowana: 16GB;
- System: Android 8.0 Oreo.

5. Eksperyment

5.1. Typy danych

Tabela 1 przedstawia zestawienie typów danych dostępnych w badanych mobilnych bazach danych.

Tabela 1: Dostępne typy danych w poszczególnych bazach

Grupa typów	SQLite	Realm	ObjectBox	SnappyDB
Typy znakowe	TEXT	STRING	STRING CHARACTER	STRING
Typy numeryczne	INTEGER REAL	INT LONG* FLOAT DOUBLE	INTEGER SHORT LONG FLOAT DOUBLE BYTE	INT SHORT LONG DOUBLE FLOAT
Typy daty	-	DATE	DATE	-
Typy logiczne	-	BOOLEAN	BOOLEAN	BOOLEAN
Typy binarne	BLOB	-	-	-
Inne	NULL	REALMMODEL REALMLIST	-	-

SQLite, w przeciwieństwie do baz obiektowych, nie posiada typu logicznego boolean. Zamiast tego wartości logiczne są przechowywane jako liczby całkowite 0 (fałsz) i 1 (prawda). Podobnie jak w przypadku boolean, SQLite nie ma zarezerwowanej klasy pamięci do przechowywania dat. Zamiast tego wbudowane funkcje SQLite mogą przechowywać daty i godziny jako wartości TEXT, REAL lub INTEGER [5,6].

Realm, poza powszechnie wykorzystywanymi typami, daje do dyspozycji także nowatorskie rozwiązania. Innowacyjnie przedstawia się tutaj możliwość przechowywania listy - RealmList <T>, gdzie T może być następującymi typami: String, Integer, Boolean, Float, Double, Short, Long, Byte, byte[] oraz Date [7].

Bazy SnappyDB oraz ObjectBox obsługują podstawowe typy danych, które nie wyróżniają się na tle innych baz.

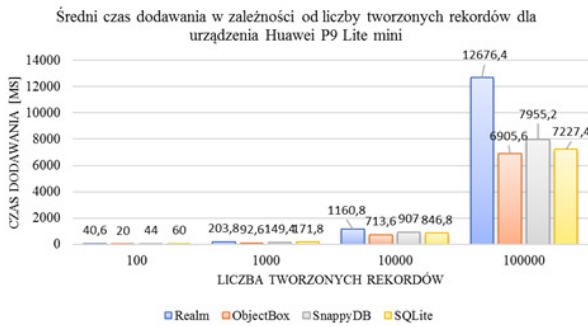
5.2. Zapis danych do bazy

Rysunek 1 przedstawia średnie czasy zapisu danych w zależności od liczby tworzonych rekordów dla urządzenia Huawei P9 Lite.

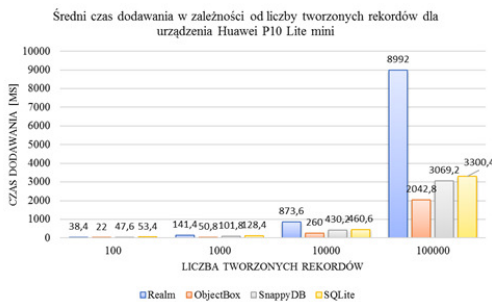
Rysunek 2 przedstawia średnie czasy zapisu danych w zależności od liczby tworzonych rekordów dla urządzenia Huawei P10 Lite mini.

Najlepsze czasy zapisywania danych na obu urządzeniach zanotowała baza ObjectBox. Najgorsze wyniki osiągnęła baza obiektowa Realm. Wszystkie badane bazy prezentują porównywalne wyniki, gdy zapisywane jest mniej niż 1000 rekordów, różnica czasu zapisu między najgorszą, a najlepszą bazą wynosi zaledwie 0,1

sekundy. W codziennym użytkowaniu aplikacji jest to nieodczuwalne.



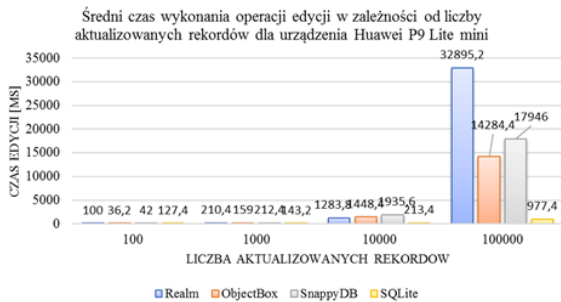
Rysunek 1: Średni czas dodawania w zależności od liczby tworzonych rekordów dla urządzenia Huawei P9 lite mini



Rysunek 2: Średni czas dodawania w zależności od liczby tworzonych rekordów dla urządzenia Huawei P10 lite mini

5.3. Edycja danych

Na rysunku 3 przedstawiony został średni czas wykonania operacji edycji w zależności od liczby aktualizowanych rekordów dla urządzenia Huawei P9 Lite mini.

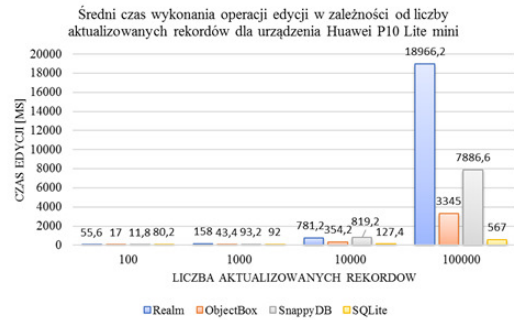


Rysunek 3: Średni czas wykonania operacji edycji w zależności od liczby aktualizowanych rekordów dla urządzenia Huawei P9 Lite mini

Na rysunku 4 przedstawiony został średni czas wykonywania operacji edycji w zależności od liczby aktualizowanych rekordów dla urządzenia Huawei P10 Lite mini.

W drugim przeprowadzonym badaniu najlepsze czasy edycji na obu urządzeniach zanotowała baza SQLite. W przypadku aktualizacji 100000 rekordów na urządzeniu Huawei P10 Lite, czas edycji w bazie SQLite był 33 razy szybszy w porównaniu do najwolniejszej bazy Realm oraz 5,9 razy szybszy w porównaniu do drugiej co do prędkości edycji bazy ObjectBox. Podobnie jak w przypadku zapisu danych różnice czasów edycji do 1000 rekordów nie przekraczają 0,1 sekundy. Analo-

gicznie ma się sytuacja między urządzeniami, w przypadku edycji do 1000 rekordów, czasy są niemalże identycznie i różnice między prędkościami poszczególnych baz nie przekraczają 50 ms.



Rysunek 4: Średni czas wykonania operacji edycji w zależności od liczby aktualizowanych rekordów dla urządzenia Huawei P10 Lite mini

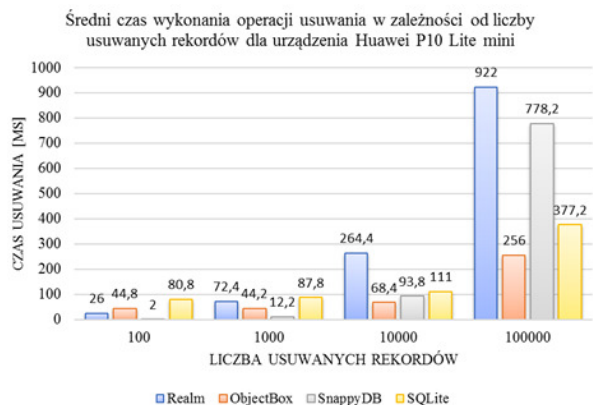
5.4. Usuwanie danych

Rysunek 5 przedstawia średnie czasy wykonania operacji usuwania w zależności od liczby rekordów dla urządzenia Huawei P9 Lite mini.



Rysunek 5: Średni czas wykonania operacji usuwania w zależności od liczby usuwanych rekordów dla urządzenia Huawei P9 Lite mini

Rysunek 6 przedstawia średnie czasy wykonania operacji usuwania w zależności od liczby rekordów dla urządzenia Huawei P10 Lite mini.



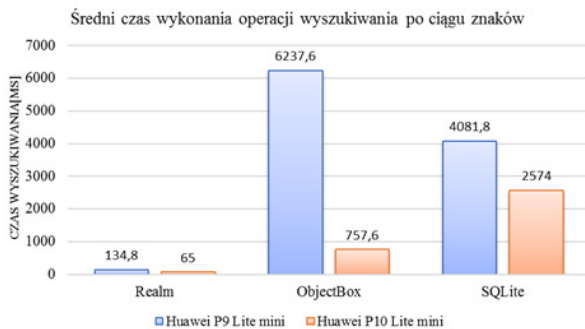
Rysunek 6: Średni czas wykonania operacji usuwania w zależności od liczby usuwanych rekordów dla urządzenia Huawei P10 Lite mini

Najlepsze czasy usuwania na obu urządzeniach osiągnęła baza obiektowa ObjectBox. Najgorsze wyniki ponownie zanotowała baza Realm. Poniżej 1000 usu-

wanych rekordów najdłuższe czasy odczytu uzyskuje SQLite. SnappyDB ma najmniejsze czasy usuwania, gdy usuwane jest mniej niż 1000 rekordów. Baza ta potrafi usunąć 100 rekordów w czasie 2 milisekund, niestety jej wydajność znacząco spada w przypadku usuwania liczby rekordów większej niż 1000. Gdy liczba rekordów jest mniejsza niż 1000, różnica w czasach usuwania badanych baz nie przekracza 0,1 sekundy.

5.5. Wyszukiwanie

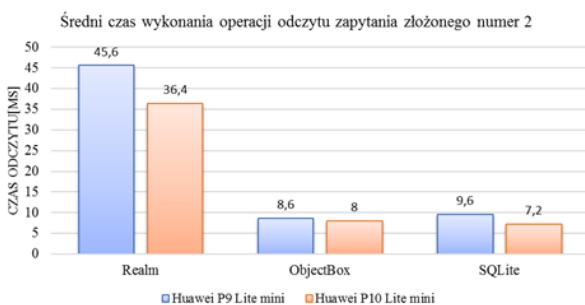
Rysunek 7 przedstawia średnie czasy wykonania operacji wyszukiwania po ciągu znaków.



Rysunek 7: Średni czas wykonania operacji wyszukiwania po ciągu znaków

Najlepsze czasy wyszukiwania danych po ciągu znaku na obu urządzeniach notuje obiektowa baza Realm. Na urządzeniu P10 Lite wyszukuje ona dane 11,6 razy szybciej niż baza ObjectBox oraz 39,6 razy szybciej niż SQLite.

Rysunek 8 przedstawia średni czas wykonania operacji odczytu poniższego zapytania złożonego. `SELECT * FROM users WHERE age > 10 AND CONTAINS(country, "Ja") LIMIT 10;`

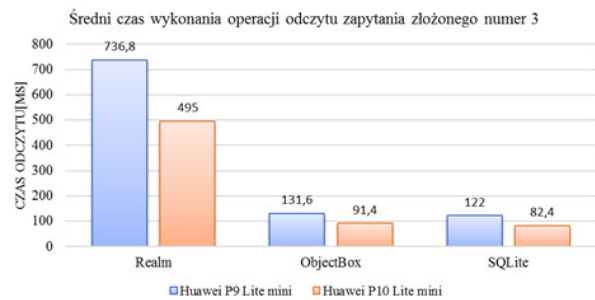


Rysunek 8: Średni czas wykonania operacji odczytu zapytania złożonego numer 2

Rysunek 9 przedstawia średni czas wykonania operacji odczytu poniższego zapytania złożonego: `SELECT DISTINCT country FROM users WHERE weight > 50 AND height < 180;`

Najlepszym wynikiem w przypadku wykonania zapytań złożonych może poszczycić się baza SQLite, niemalże na równi z bazą ObjectBox. W przypadku tych dwóch baz czas odczytu złożonego zapytania numer 2 nie przekraczał 10 milisekund na obu urządzeniach. Najgorsze wyniki zanotowała baza Realm, w przypadku zapytania drugiego notująca na urządzeniu Huawei P9

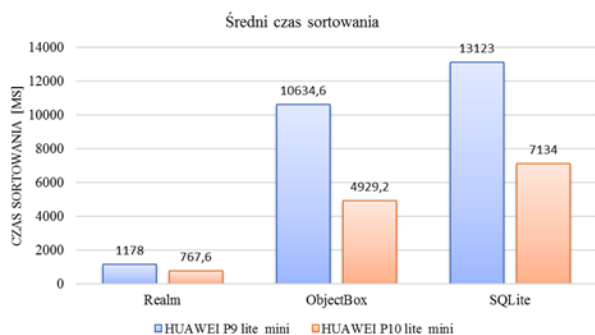
Lite mini czasy 5,3 razy gorsze od najszybszej w tym teście bazy ObjectBox.



Rysunek 9: Średni czas wykonania operacji odczytu zapytania złożonego numer 3

5.6. Sortowanie

Rysunek 10 przedstawia średni czas sortowania danych.

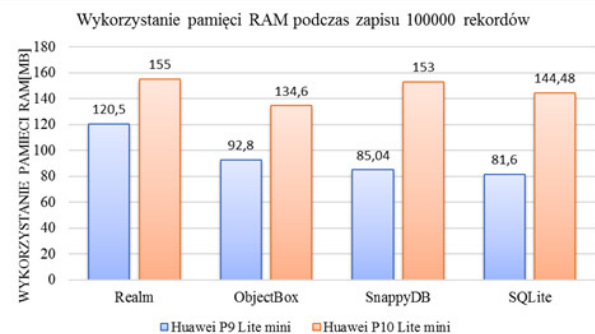


Rysunek 10: Średni czas sortowania wieku użytkowników od najmniejszego do największego.

Najlepsze czasy sortowania rekordów uzyskuje baza Realm. W przypadku operacji wykonanych na telefonie Huawei P10 Lite mini, czasy sortowania bazy Realm są 6,4 razy krótsze niż czasu konkurencyjnej, obiektowej bazy ObjectBox. Najgorzej w tym teście wypadła baza SQLite, notująca najdłuższe czasy wykonania operacji sortowania. Różnica między najlepszą, a najgorszą bazą testowaną na urządzeniu P9 Lite wynosi aż 11,94 sekundy. Różnica ta jest bardzo duża i byłaby wyczuwalna podczas normalnego użytkownika aplikacji.

5.7. Wykorzystywanie zasobów

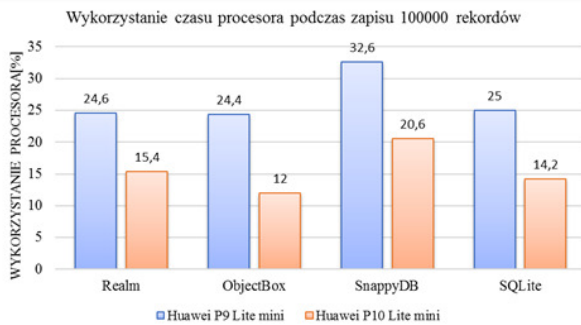
Rysunek 11 przedstawia wykorzystanie pamięci RAM podczas zapisu 100000 rekordów.



Rysunek 11: Średnie wykorzystanie pamięci RAM podczas zapisu 100000 rekordów

Największe zużycie pamięci RAM powoduje Realm. W przypadku badania wykonanego na telefonie Huawei P10 Lite, badane bazy prezentują niemalże identyczne zużycie pamięci RAM. Podczas operacji zapisu, różnica między największym, a najmniejszym zużyciem wynosi 20MB. Wykorzystanie przez bazę Realm 155MB pamięci RAM stanowi zaledwie 4% pamięci RAM posiadanej przez model P10 Lite.

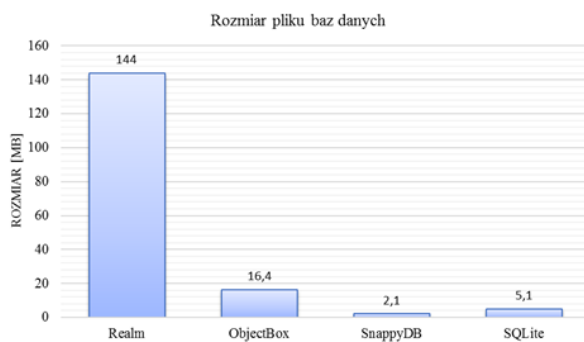
Rysunek 12 przedstawia wykorzystanie czasu procesora podczas zapisu 100000 rekordów.



Rysunek 12: Średnie wykorzystanie czasu procesora podczas zapisu 100000 rekordów

Podczas operacji zapisu największe obciążenie procesora na obu urządzeniach generuje baza SnappyDB. Pozostałe testowane bazy wykorzystują procesor w podobny sposób. W przypadku pracy na urządzeniu Huawei P9 Lite, zużycie procesora w czasie wykonania operacji zapisu jest niemalże identyczne dla baz SQLite, Realm oraz ObjectBox i wynosi około 25%. Dla porównania SnappyDB wykonując tę samą operację na tym samym telefonie obciąża procesor aż w 33%.

Rysunek 13 przedstawia rozmiary plików baz danych znajdujących się na dysku wewnętrznym.



Rysunek 13: Rozmiar pliku bazy danych

Na podstawie przeprowadzonego badania można obserwować, że rozmiar pliku bazy Realm jest kilkukrotnie większy od pozostałych. Najmniej miejsca na dysku zajmuje prosta baza SnappyDB, która spośród testowanych baz posiada najmniej funkcjonalności. W przypadku zapisanych 100000 rekordów baza SQLite zajmuje jedynie 5,1MB. Według dokumentacji, maksymalny rozmiar bazy danych może wynosić około $1,4E14$ bajtów (140 terabajtów) [5].

6. Wnioski

Celem niniejszego artykułu było porównanie baz danych dedykowanych dla systemu Android. Na podstawie przeprowadzonych badań należy podkreślić, że nie można jednoznacznie wyznaczyć najlepszej bazy danych. Przy wyborze systemu bazodanowego, najważniejsze jest określenie indywidualnych potrzeb stawianych przez wymagania aplikacji, oraz wybór pod tym względem odpowiedniej bazy danych.

Podczas wykonywania operacji CRUD wszystkie z badanych baz wykazywały niemalże podobne czasy w przypadku pracy z liczbą rekordów mniejszą niż 1000. Podczas operacji zapisu, edycji oraz usuwania dużej liczby rekordów najgorsze czasy notowała baza Realm. Najlepiej z aktualizacją rekordów poradziła sobie relacyjna baza SQLite, wykonująca operacje edycji kilkukrotnie szybciej niż pozostałe badane rozwiązania bazodanowe. Badanie operacji wyszukiwania danych ciągu znaku wykazało, że baza Realm jest kilkunastokrotnie szybsza od bazy SQLite. W przypadku odczytu złożonych zapytań najlepsze czasy notowała baza SQLite oraz ObjectBox, mająca niewiele słabsze czasy. Operacja sortowania najszybciej wykonywana była na bazie Realm. Różnice czasu sortowania pomiędzy bazą Realm, a pozostałymi badanymi technologiami wynosiły nawet kilka sekund. Wykorzystanie zasobów takich jak procesor oraz RAM było niewielkie w stosunku do całkowitych zasobów badanych smartfonów. Największą rozbieżność można zauważyć w przypadku badania wielkości pliku bazy danych. Rozmiar pliku bazy Realm jest wielokrotnie większy od pozostałych, co w przypadku obecnie wykorzystywanych smartfonów, posiadających ponad 100GB pamięci wewnętrznej, jest mało znaczące.

Literatura

- [1] Ranking użytkowników sieci komórkowej <https://www.parkiet.com/Analizy/305159869-Liczba-kart-SIM-W-I-kwartale-2019-roku-na-rynku-telefonii-komorkowej-w-Polsce-widac-stabilizacje-baz-klientow.html> [03.11.2019]
- [2] Raport Digital 2019 <https://mobirank.pl/2019/02/01/raport-mobile-i-digital-na-swiecie-w-2019-roku/> [03.11.2019]
- [3] Sondaż CBOS na temat wykorzystania telefonów komórkowych <https://technologia.dziennik.pl/internet/artykuly/555355.sondaz-telefon-y-komrkowe-smartfony.html> [03.11.2019]
- [4] Strona sklepu Google Play <https://play.google.com/store/apps?hl=pl> [03.11.2019]
- [5] Kumar Karn Vikas, Kumar Aditya Sunny, Android SQLite Essentials, Packt Publishing, 2014
- [6] Dokumentacja SQLite <https://www.sqlite.org/docs.html> [03.11.2019]
- [7] Dokumentacja Realm <https://realm.io/docs/java/latest/> [03.11.2019]
- [8] Dokumentacja ObjectBox <https://docs.objectbox.io/getting-started> [03.11.2019]

- [9] Oficjalna strona projektu SnappyDB <https://github.com/nhachicha/SnappyDB> [03.11.2019]
- [10] T. Perrier, F. Pervaiz, NoSQL in a Mobile World Benchmarking Embedded Mobile Databases, 2013
- [11] K. Kussainov, B. Kumalakov, Mobile Data Store Platforms: Test Case based Performance Evaluation, 2016
- [12] Farzad Tavakkoli, Azam Andalib, Asadollah Shahbahrami and Reza Ebrahimi Atani, A Comparison of Lightweight Databases in Mobile Systems, 2011
- [13] M. Grudzień, K. Korgol, D. Gutek, Comparison of the possible uses and performance analysis of databases on mobile operating systems, 2016
- [14] Oficjalna strona projektu Java Faker <https://github.com/DiUS/java-faker> [03.11.2019]