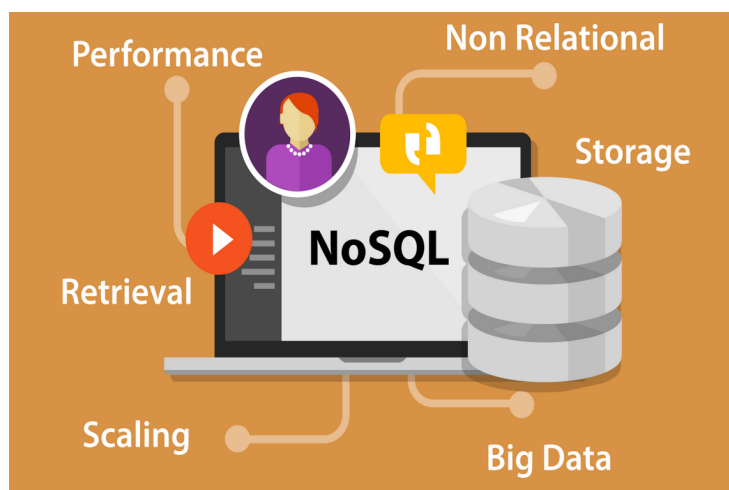




Универзитет „Св. Кирил и Методиј“ во Скопје  
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И  
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

# Клуч-вредност бази на податоци

Предмет: Неструктурирани бази на податоци



Изработиле:

Ангела Мојсовска 171115  
Елеонора Закоска 171173  
Емилија Радоешка 171028  
Тамара Таневска 171082

Ментор:

Д-р Слободан Калајџиски

Септември, 2021

# СОДРЖИНА

<b>1. Вовед</b>	<b>3</b>
1.1 Разлики помеѓу SQL и NoSQL базите на податоци	3
1.2 NoSQL - Key value	4
1.2.1 Redis	6
1.3 SQL - MySQL	7
<b>2. Анализа на податоци</b>	<b>7</b>
2.1 Опис на множеството податоци	7
<b>3. Внесување податоци во базите</b>	<b>9</b>
3.1 Внесување во Redis	9
3.1.1 Внесување рачно редица по редица	9
3.1.2 Внесување со импортирање од JSON	11
3.2 Внесување во MySQL	12
<b>4. Прашалници и споредба помеѓу Redis и MySQL</b>	<b>14</b>
4.1 Прашалници во Redis	14
4.2 Прашалници во MySQL	15
4.3 Споредба	18
<b>5. Заклучок</b>	<b>19</b>
<b>6. Користена литература</b>	<b>20</b>

# 1. Вовед

Користење на социјалните мрежи и медиуми, без разлика дали се некои од водечките компании како Google, Facebook, Instagram, Twitter или некоја помала компанија во која внесуваме податоци, секојдневно собираат голем број податоци. Овие апликации генерираат огромни количини на неструктурирани податоци, кои некогаш знаат да бидат предизвик за обработка. Податоците кои се добиваат се изворот на сите направени анализи, евалуации, позадина за донесување одлуки и поради нивната улога важно е да бидат зачувани во соодветни структури. Релациони бази на податоци се еден начин на зачувување на податоци, но поради нивната ограниченост за работа, само со структурирани податоци, се појавува потребата за NoSQL Database менаџмент системи.

Во нашиот проект ќе зборуваме за предностите на NoSQL базите на податоци, различните видови и подетално ќе се задржиме на key-value NoSQL бази на податоци.

## 1.1 Разлики помеѓу SQL и NoSQL базите на податоци

Релационите бази се докажан модел кога станува збор за чување на податоци. Тие гарантираат ефикасност, конзистентност, сигурност преку користењето на атомични, изолирани и постојани трансакции. Но, и покрај тоа што овие бази се доминантна технологија за чување на податоци, нивните стриктни правила и мерки ги прават несоодветни за складирање на секојдневните податоци. Овие недостатоци ги дополнуваат NoSQL базите на податоци.

Еден од главните проблеми е зголемениот број на податоци кои е потребно да се генерираат и зачувуваат. Релационите бази, иако за помала количина на податоци успеваат да ги извршат прашањата во прифатливо време, кога станува збор за големи множества на податоци времето на извршување е многу поголемо и некорисно.

Друг проблем кој се појавува доколку се користат релациони бази за чување, тие податоци треба да се структурирани по некоја шема и правила, а тоа најчесто не е случај со секојдневните т.н big data податоци. Овие податоци најчесто се непотполни, рапидно се зголемуваат и со тоа и рапидно се менуваат како и нивните шеми за кои не би можеле да дадеме никаква претпоставка.

Ваквите проблеми налагаат користење на нов тип на бази на податоци наречени NoSQL бази на податоци. NoSQL базите на податоци одговараат на недостатоците на веќе постоечките и горенаведените релациони бази на податоци и промовираат перформанси, флексибилност и скалабилност кои се потребни за модерните апликации. Помеѓу 3 различните видови на неструктурирани бази на податоци има многу разлики и многу мала

сличност настрана од тоа што не користат релационен модел. Тие можат да се поделат на четири типа:

- **Key-value** – секој запис во базата е зачуван како име на атрибут ( или клуч ) заедно со неговата вредност. {Пример- Riak, Amazon S3 (Dynamo)}

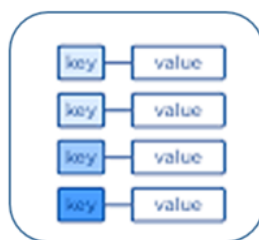
- **Document** - бази каде што секој клуч е во пар со сложена податочна структура наречена документ. Документите можат да содржат многу различни key-value, keyarray парови или вгнездени документи. {Пример- MongoDB, CouchDB}

- **Column family** – бази како Cassandra и HBase кои се оптимизирани за прашања врз големи податочни множества и наместо редици, чуваат колони од податоци. {Пример- HBase, Cassandra}

- **Graph** – бази кои се користат за зачувување на информации за мрежи од податоци, како социјални врски (Neo4J, Giraph). {Пример- Neo4J}



Document Store



Key-Value Store



Wide-Column Store



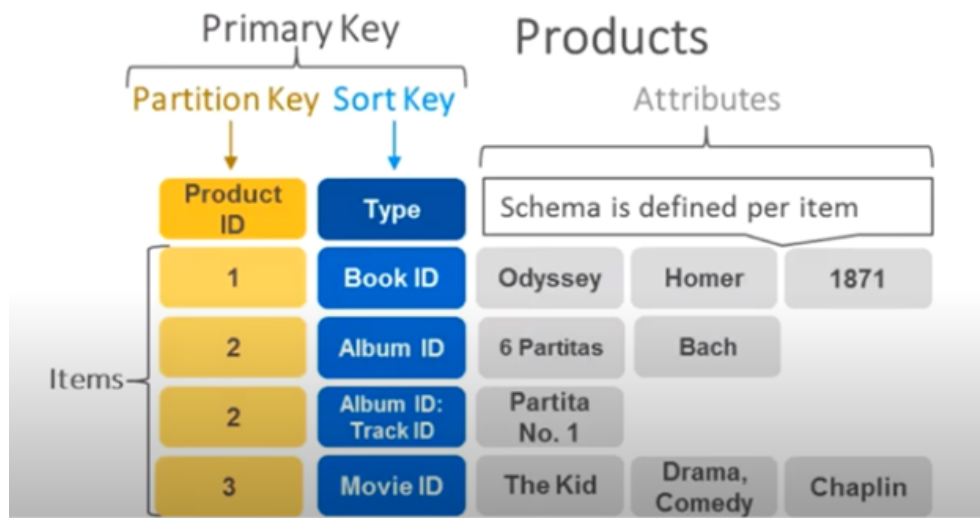
Graph Store

## 1.2 NoSQL - Key value

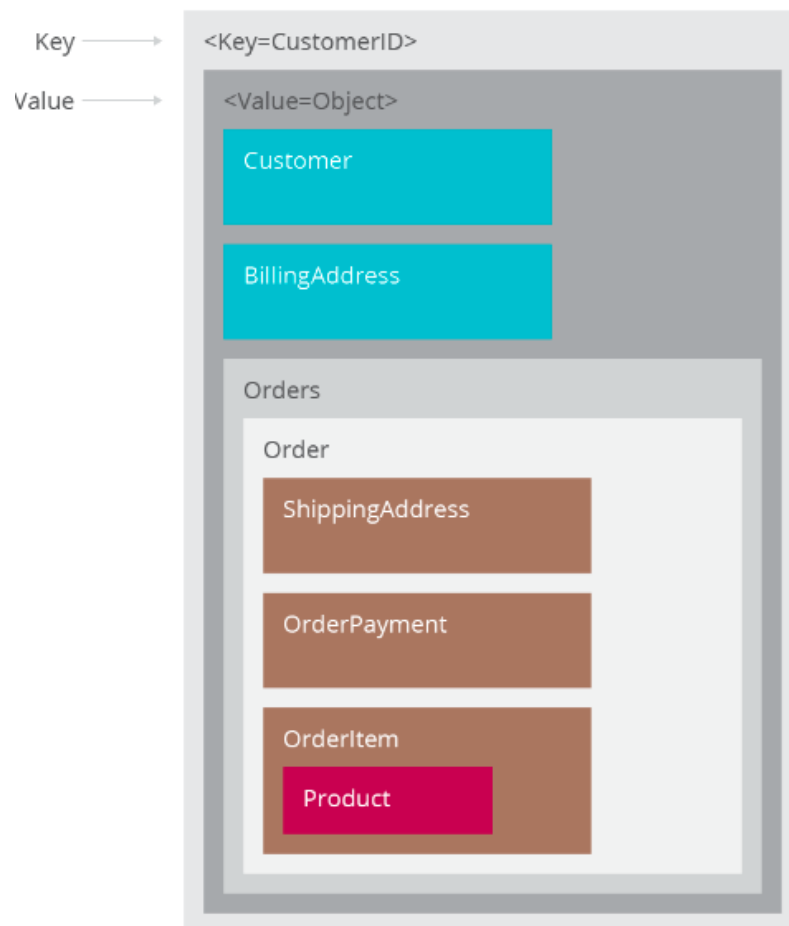
Key-value базите на податоци користи едноставен механизам со клуч и вредност со цел да зачува запис. Овие бази имаат внатрешен кеш механизам кои им овозможува брзина и ефикасност. Најчесто се поставуваат повеќе сервери, бидејќи нивното посатавување е доста едноставно. Затоа овозможуваат најлесно партиционирање и хоризонтална скалабилност од сите NoSQL бази.

Како работи механизмот на клуч и вредност?

Место да има табели и записи има парови на клуч и вредности, каде што клучот е единствен идентификатор (и клучот и вредноста може да бидат било што од едноставни до комплексни објекти). Вредностите може да имаат било каква структура и тука нема никаква шема. Секој пар покрај клучот има и покажувач, а тоа овозможува брз пристап до податоците. Не мора сите записи да имаат иста структура.



Пример .За продукт со ID 1 , и тип BOOK ID (како примарен клуч), има име на книгата авторот и година.



Најважни карактеристики:

**Складирање и влечење податоци** – користејќи го клучот се овозможува брз пристап до складираните податоци. Механизам што се користи е правење на индекс по клучната колона.

**Кеширање** – дополнително за побрз пристап имаат интегриран механизам за кеширање. Најчесто користените информации се ставаат во кеш. При кластерот сервери, серверите може да комуницираат меѓу себе за да споделуваат податоци.

**Едноставна структура на податоците**- Самата структура овозможува брз пристап до податоците, имаа проста структура на податоците

**Скалабилност**- лесно се додаваат и одземаат сервери во зависност од потребите.

## 1.2.1 Redis

Redis е моментално најпознатата key-value база на податоци. Open - source во in-memory data store која може да се користи и како база и/или како кеш и message broker. Во Redis се е на исто ниво, не постои некоја хиерархија и флексибилен е односно нема дефинирање на шеми ниту пак типови на колони. Податочни типови кои ги поддржува се: Strings, lists, sets, sorted sets. Hashes, bitmap, hyperlogs, geospatial indexes.

Редис има вградено кеширање и е многу брз, што следува од тоа дека е база за клуч вредност тип, па така може да прави 110 000(sets) внесувања во секунда и 81000(gets) читање во секунда.



## 1.3 SQL - MySQL

MySQL е систем за управување со релациона база на податоци со отворен код. Како и секоја друга SQL база, податоците со чуваат во табели и се структурирани. MySQL е една од најкористените бази за структурирани податоци, дозволува додавање, ажурирање, бришење, менување и уште многу опции. Дополнително брз е, доверлив, податоците ни се безбедни, има високи перформанси и е флексибилен . Најчесто се користи за веб апликации.



## 2. Анализа на податоци

Податоците кои се користат се множество на мобилни телефони со нивните перформанси како колони. Нашиот проект е анализирање на две бази и како тие реагираат на едно истото податочно множество, односно споредување на перформансите на двете бази.

Податоците содржат повеќе csv. датотеки кои сакаме да ги искористиме. Се спојуваат својствата од повеќето датотеки во една merged.csv , следно треба да се внесат и во двете бази, за на крај да се споредат времето на извршување на истите и да се донесе заклучок која е побрза.

### 2.1 Опис на множеството податоци

Избраното множество на податоци е множество на мобилни телефони со нивните перформанси. Замислата на проектот е онлајн продавница за мобилни телефони каде што базата во позадина која ја користи продавницата е нашата неструктурирана база merged.csv. Идејата е вработените, но и клиентите да можат да ги разгледуваат или пребаруваат телефоните според некој избран перформанс.

Податочното множество ги има следните својства:

- Id - ид на продуктот
- Brand - од кој производител е
- Battery saracity- колкав е капацитетот на батеријата
- Screen size- колкав е екранот
- Touchscreen - дали е на допир
- Internal storage- меморијата колкава е
- Headphones- дали има слушалки
- Face unlock- дали се отвара со препознавање на ликот
- Fingerprint sensor - дали се отвара со сензор на прстот
- Wireless charging- дали може да се полни без полнач
- Number of SIMs- колку сим картички поддржува
- SIM Type - каква сим има

	A	B	C	D	E	F	G	H	I	J	K	L
1	id	brand	battery	screen size	touch	storage	headphones	face unlock	fingerprint	wifi charge	no. sims	simtype
2	378	micromax	3000	5,5	yes	32	yes	no	yes	no	1.0	nano

И уште некои својства од целосното множество merge.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W																										
1	Brand,Battery capacity,Screen Size,Touchscreen,Internal storage,Headphones,Face unlock,Fingerprint sensor,Wireless charging,NFC,Number of SIMs,SIM Type,4G/ LTE,5G,Expandable storage,Price,3G,EDGE,GPRS,Wi-Fi,VoLTE,Usb Type,Bluetooth,OS,cpuCore																																																
2	0	Op	4000	6.53	Yes	128	Yes	Yes	No	2.0	Nano	No	256	23990	Yes	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,mediatek	2.1	Yes	8	No	48	LED	16	394	75.8	161.8	195	Yes	No	Yes	August	2019	4	1080	2340	1.1	1.1	0.0	0.1	1.0	0	
3	1	Apple	1960	4.7	Yes	32	Yes	No	Yes	1.0	Nano	Yes	No	0	27499	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Quad,apple	2.34	Yes	2	No	12	LED	7	326	67.09	138.3	138	No	No	September	2016	1	750	1334	1.1	1.1	0.0	0.1	1.0	0		
4	2	Apple	1960	4.7	Yes	32	Yes	No	Yes	1.0	Nano	Yes	No	0	19890	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Quad,apple	2.34	Yes	2	No	12	LED	7	326	67.09	138.3	138	No	No	September	2016	1	750	1334	1.1	1.1	0.0	0.1	1.0	0		
5	3	Apple	1715	4.7	Yes	16	Yes	No	Yes	1.0	Nano	Yes	No	0	41999	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Dual,apple	1.84	Yes	2	No	12	Dual LED	5	326	67.1	138.3	143	No	No	September	2015	1	750	1334	1.1	1.1	0.0	0.1	1.0	0		
6	4	Apple	1715	4.7	Yes	16	Yes	No	Yes	1.0	Nano	Yes	No	0	23498	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Dual,apple	1.84	Yes	2	No	12	Dual LED	5	326	67.1	138.3	143	No	No	September	2015	1	750	1334	1.1	1.1	0.0	0.1	1.0	0		
7	5	Apple	1715	4.7	Yes	32	Yes	No	Yes	1.0	Nano	Yes	No	0	41999	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Dual,apple	1.84	Yes	2	No	12	Dual LED	5	326	67.1	138.3	143	No	No	September	2015	1	750	1334	1.1	1.1	0.0	0.1	1.0	0		
8	6	Apple	1715	4.7	Yes	32	Yes	No	Yes	1.0	Nano	Yes	No	0	23498	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Dual,apple	1.84	Yes	2	No	12	Dual LED	5	326	67.1	138.3	143	No	No	September	2015	1	750	1334	1.1	1.1	0.0	0.1	1.0	0		
9	7	Apple	1715	4.7	Yes	64	Yes	No	Yes	1.0	Nano	Yes	No	0	41999	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Dual,apple	1.84	Yes	2	No	12	Dual LED	5	326	67.1	138.3	143	No	No	September	2015	1	750	1334	1.1	1.1	0.0	0.1	1.0	0		
10	8	Apple	1715	4.7	Yes	64	Yes	No	Yes	1.0	Nano	Yes	No	0	23498	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Dual,apple	1.84	Yes	2	No	12	Dual LED	5	326	67.1	138.3	143	No	No	September	2015	1	750	1334	1.1	1.1	0.0	0.1	1.0	0		
11	9	Vivo	5000	6.35	Yes	64	Yes	Yes	No	2.0	Nano	Yes	No	256	10990	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,mediatek	2	Yes	4	Yes	13	LED	8	268	76.8	159.4	190.5	No	No	Yes	May	2019	3	720	1544	1.1	1.1	0.0	0.0	1.0	0	
12	10	Vivo	5000	6.35	Yes	64	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	9885	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,mediatek	2	Yes	3	Yes	13	LED	8	268	76.8	159.4	190.5	No	No	Yes	May	2019	3	720	1544	1.1	1.1	0.0	0.0	1.0	0
13	11	Vivo	5000	6.35	Yes	32	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	10990	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,mediatek	2	Yes	4	Yes	13	LED	8	268	76.8	159.4	190.5	No	No	Yes	May	2019	3	720	1544	1.1	1.1	0.0	0.0	1.0	0
14	12	Vivo	5000	6.35	Yes	32	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	9885	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,mediatek	2	Yes	4	Yes	13	LED	8	268	76.8	159.4	190.5	No	No	Yes	May	2019	3	720	1544	1.1	1.1	0.0	0.0	1.0	0
15	13	Op	5000	6.5	Yes	128	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	15990	Yes	Yes	Yes	Yes	USB-C	Yes	Android	Octa,qualcomm	2	Yes	4	Yes	48	Dual LED	16	270	75.6	163.6	195	No	No	Yes	September	2019	4	720	1600	1.1	1.1	0.0	0.1	1.0	0
16	14	Op	5000	6.5	Yes	128	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	15990	Yes	Yes	Yes	Yes	USB-C	Yes	Android	Octa,qualcomm	2	Yes	4	Yes	48	Dual LED	16	270	75.6	163.6	195	No	No	Yes	September	2019	4	720	1600	1.1	1.1	0.0	0.1	1.0	0
17	15	Op	5000	6.5	Yes	64	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	12990	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,qualcomm	2	Yes	4	Yes	12	LED	8	270	75.6	163.6	195	No	No	Yes	September	2019	4	720	1600	1.0	1.1	0.0	0.1	1.0	0
18	16	Op	5000	6.5	Yes	64	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	12990	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,qualcomm	2	Yes	4	Yes	12	LED	8	270	75.6	163.6	195	No	No	Yes	September	2019	4	720	1600	1.0	1.1	0.0	0.1	1.0	0
19	17	Vivo	4500	6.38	Yes	128	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	19990	Yes	Yes	Yes	Yes	USB-C	Yes	Android	Octa,qualcomm	2	Yes	8	Yes	48	LED	32	404	75.19	159.25	186.7	Yes	No	Yes	November	2019	4	1080	2340	1.1	1.1	0.0	0.1	1.0	0
20	18	Vivo	4030	6.22	Yes	32	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	8490	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,mediatek	2	Yes	3	Yes	13	LED	8	270	75.1	155.1	163.5	No	No	Yes	January	2019	2	720	1520	1.1	1.1	0.0	0.0	1.0	0
21	19	Vivo	4030	6.22	Yes	32	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	8490	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,mediatek	2	Yes	2	Yes	13	LED	8	270	75.1	155.1	163.5	No	No	Yes	January	2019	2	720	1520	1.1	1.1	0.0	0.0	1.0	0
22	20	Vivo	4030	6.22	Yes	32	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	8490	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,mediatek	2	Yes	3	Yes	13	LED	8	270	75.1	155.1	163.5	No	No	Yes	January	2019	2	720	1520	1.1	1.1	0.0	0.0	1.0	0
23	21	Vivo	4030	6.22	Yes	32	Yes	Yes	Yes	No	2.0	Nano	Yes	No	256	8490	Yes	Yes	Yes	Yes	microUSB	Yes	Android	Octa,mediatek	2	Yes	2	Yes	13	LED	8	270	75.1	155.1	163.5	No	No	Yes	January	2019	2	720	1520	1.1	1.1	0.0	0.0	1.0	0
24	22	Apple	2942	6.1	Yes	64	Yes	Yes	No	Yes	2.0	eSIM	Yes	No	0	42900	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Hexa,apple	2.49	Yes	3	No	12	LED	7	324	75.7	150.9	194	Yes	No	September	2018	1	1792	828	1.1	1.1	0.0	0.1	1.0	0	
25	Apple	2942	6.1	Yes	128	Yes	Yes	No	Yes	2.0	eSIM	Yes	No	0	51799	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Hexa,apple	2.49	Yes	3	No	12	LED	7	324	75.7	150.9	194	Yes	No	September	2018	1	1792	828	1.1	1.1	0.0	0.1	1.0	0		
26	Apple	2942	6.1	Yes	64	Yes	Yes	No	Yes	2.0	eSIM	Yes	No	0	42900	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Hexa,apple	2.49	Yes	3	No	12	LED	7	324	75.7	150.9	194	Yes	No	September	2018	1	1792	828	1.1	1.1	0.0	0.1	1.0	0		
27	Apple	2942	6.1	Yes	128	Yes	Yes	No	Yes	2.0	eSIM	Yes	No	0	51799	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Hexa,apple	2.49	Yes	3	No	12	LED	7	324	75.7	150.9	194	Yes	No	September	2018	1	1792	828	1.1	1.1	0.0	0.1	1.0	0		
28	Apple	2942	6.1	Yes	64	Yes	Yes	No	Yes	2.0	eSIM	Yes	No	0	42900	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Hexa,apple	2.49	Yes	3	No	12	LED	7	324	75.7	150.9	194	Yes	No	September	2018	1	1792	828	1.1	1.1	0.0	0.1	1.0	0		
29	Apple	2942	6.1	Yes	128	Yes	Yes	No	Yes	2.0	eSIM	Yes	No	0	51799	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Hexa,apple	2.49	Yes	3	No	12	LED	7	324	75.7	150.9	194	Yes	No	September	2018	1	1792	828	1.1	1.1	0.0	0.1	1.0	0		
30	Apple	2942	6.1	Yes	64	Yes	Yes	No	Yes	2.0	eSIM	Yes	No	0	42900	Yes	Yes	Yes	Yes	Proprietary	Yes	iOS	Hexa,apple	2.49	Yes	3	No	12	LED	7	324	75.7	150.9	194	Yes	No	September	2018	1	1792	828	1.1	1.1	0.0	0.1	1.0	0		



## 3. Внесување податоци во базите

### 3.1 Внесување во Redis

#### 3.1.1 Внесување рачно редица по редица

За поврзување со базата Redis , се поврзува преку cmd со датотеката redis-server.exe на port 6379 , а командите се извршуваат во Windows Power Shell со стартување на датотеката redis-cli.exe која се поврзува со локалниот сервер (нашиот компјутер).

```
C:\Users\Eleonora Zakoska\Desktop>cd redis
C:\Users\Eleonora Zakoska\Desktop\redis>redis-server.exe
[1900] 17 Sep 16:44:37.249 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server.exe /path/to/redis.conf

Redis 3.0.504 (00000000/0) 64 bit
Running in standalone mode
Port: 6379
PID: 1900

http://redis.io

[1900] 17 Sep 16:44:37.249 # Server started, Redis version 3.0.504
[1900] 17 Sep 16:44:37.249 * The server is now ready to accept connections on port 6379
[1900] 17 Sep 19:27:45.231 * 1 changes in 3600 seconds. Saving...
[1900] 17 Sep 19:27:45.231 * Background saving started by pid 3840
[1900] 17 Sep 19:27:45.451 # fork operation complete
[1900] 17 Sep 19:27:45.451 * Background saving terminated with success
```

За да се внесат податоци во Redis, најлесниот и најправилен начин за внесување е хеш табела. Со помош на hesh табелите се избегнува редудантност, односно , со клучот може да пристапиме до сите информации.

Се чува листа со сите продукти во продавницата (а за секој продукт има својства). Секој продукт е наречен **prod** , а продуктите се чуваат во **phones**.

```

127.0.0.1:6379> hmset phones:prod1 uniqueid "0" brand "Oppo" battery "4000" screensize "6.53" touch "yes" storage "128"
OK
127.0.0.1:6379> hmset phones:prod1 headphones "yes" faceun "yes" fingerprint "yes" wcharge "no" nosims "2.0" simtype "nano"
OK
127.0.0.1:6379> hget phones:prod1
(error) ERR wrong number of arguments for 'hget' command
127.0.0.1:6379> hgetall phones:prod1
1) "uniqueid"
2) "0"
3) "brand"
4) "Oppo"
5) "battery"
6) "4000"
7) "screensize"
8) "6.53"
9) "touch"
10) "yes"
11) "storage"
12) "128"
13) "headphones"
14) "yes"
15) "faceun"
16) "yes"
17) "fingerprint"
18) "yes"
19) "wcharge"
20) "no"
21) "nosims"
22) "2.0"
23) "simtype"
24) "nano"

```

Првиот продукт е поставен со командата **hmset phones:prod 1 (key “value” ....** Колку и да ни е потребно вакви парови на својства). Со командата **hgetall phones:prod1** може да се увиди тоа.

Може и да се бришат продукти со командата **hdel**, затоа еве пример како може да се избрише некое својство на некој продукт.

```

127.0.0.1:6379> hdel phones:prod1 brand
(integer) 1
127.0.0.1:6379>

```

Или пак може да се пребаруваат податоци по клуч или по вредности. **Hkeys** пребарува по клучеви за даден продукт, а **hvals** пребарува по вредности за даден продукт.

```

127.0.0.1:6379> hkeys phones:prod378
1) "uniqueid"
2) "brand"
3) "battery"
4) "screensize"
5) "touch"
6) "storage"
7) "headphones"
8) "faceun"
9) "fingereprint"
10) "wcharge"
11) "nosims"
12) "simtype"
127.0.0.1:6379> hvals phones:prod1
1) "0"
2) "Oppo"
3) "4000"
4) "6.53"
5) "yes"
6) "128"
7) "yes"
8) "yes"
9) "yes"
10) "no"
11) "2.0"
12) "nano"
127.0.0.1:6379>

```

### 3.1.2 Внесување со импортирање од JSON

Податоците покрај csv датотека постојат и како JSON датотека и истите се прикажуваат во Redis користејќи PyCharm.

Пред се преку python console се внесуваат потребните библиотеки.

```
$ pip install redis
```

Потоа од постоечката JSON датотека се превземаат сите податоци и се претставуваат во key-value база со помош на следниот код.

```
1  import json
2  from pprint import pprint
3
4  import redis
5  import time
6
7
8
9  f = open('C:/Users/MARKO/Downloads/mobile_phones.json', "r")
10 r = redis.StrictRedis(host='localhost', port=6379, db=1)
11 data = json.loads(f.read())
12 start_time = time.time()
13 cnt = 0
14 for key in data:
15     for value in key:
16         r.hset("test_json", str(key), str(value))
17         cnt += 1
18
19 pprint(r.hgetall('test_json'))
20 print("--- %s seconds ---" % (time.time() - start_time))
21 print(cnt)
22
23
```

Приказот на податоците по внесот, може да се испечати и прикаже на екран, а тоа е следниот излез.

```
19 pprint(r.hgetall('test_json'))
20 print("--- %s seconds ---" % (time.time() - start_time))
21 print(cnt)
```

Run: main x

```
b"{'Brand': 'Xiaomi', 'Model': 'Redmi 4', 'Launched': 'November 2016', 'Launched in India': 'Yes', 'Fo
b"{'Brand': 'Xiaomi', 'Model': 'Redmi 5', 'Launched': 'December 2017', 'Launched in India': 'Yes', 'Fo
b"{'Brand': 'Xiaomi', 'Model': 'Redmi 6 Pro', 'Launched': 'September 2018', 'Launched in India': 'Yes'
b"{'Brand': 'Xiaomi', 'Model': 'Redmi 6', 'Launched': 'June 2018', 'Launched in India': 'Yes', 'Form f
b"{'Brand': 'Xiaomi', 'Model': 'Redmi 6', 'Launched': 'June 2018', 'Launched in India': 'Yes', 'Form f
b"{'Brand': 'Xiaomi', 'Model': 'Redmi 6A', 'Launched': 'June 2018', 'Launched in India': 'Yes', 'Form
b"{'Brand': 'Xiaomi', 'Model': 'Redmi 7', 'Launched': 'March 2019', 'Launched in India': 'Yes', 'Form
```

## 3.2 Внесување во MySQL

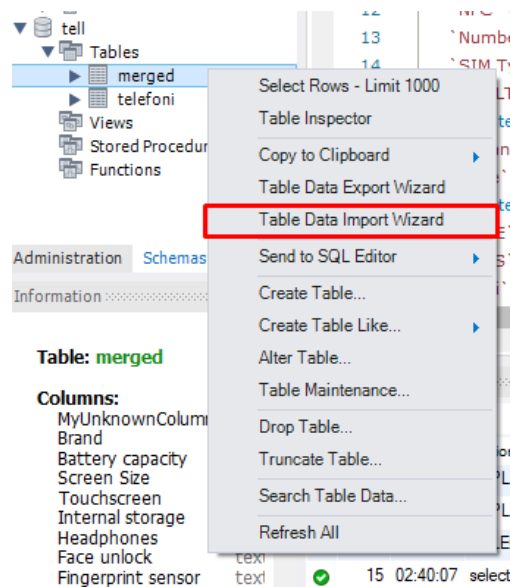
За да се внесе постоечката база на податоци од обработена csv датотека во SQL Table се користи MySQL Workbench. Прво се креира базата односно со use db\_name се наведува во која база ќе биде пристапена. Потоа се креира табела со сите атрибути како и нивните типови, во овој случај наведени се сите атрибути кои ги содржи csv датотеката.

```

1 CREATE TABLE `merged` (
2   `MyUnknownColumn` int DEFAULT NULL,
3   `Brand` text,
4   `Battery capacity` int DEFAULT NULL,
5   `Screen Size` double DEFAULT NULL,
6   `Touchscreen` text,
7   `Internal storage` int DEFAULT NULL,
8   `Headphones` text,
9   `Face unlock` text,
10  `Fingerprint sensor` text,
11  `Wireless charging` text,
12  `NFC` text,
13  `Number of SIMs` double DEFAULT NULL,
14  `SIM Type` text,
15  `4G/ LTE` text,
16  `5G` text,
17  `Expandable storage` int DEFAULT NULL,
18  `Price` int DEFAULT NULL,
19  `3G` text,
20  `EDGE` text,
21  `GPRS` text,
22  `Wi-Fi` text,

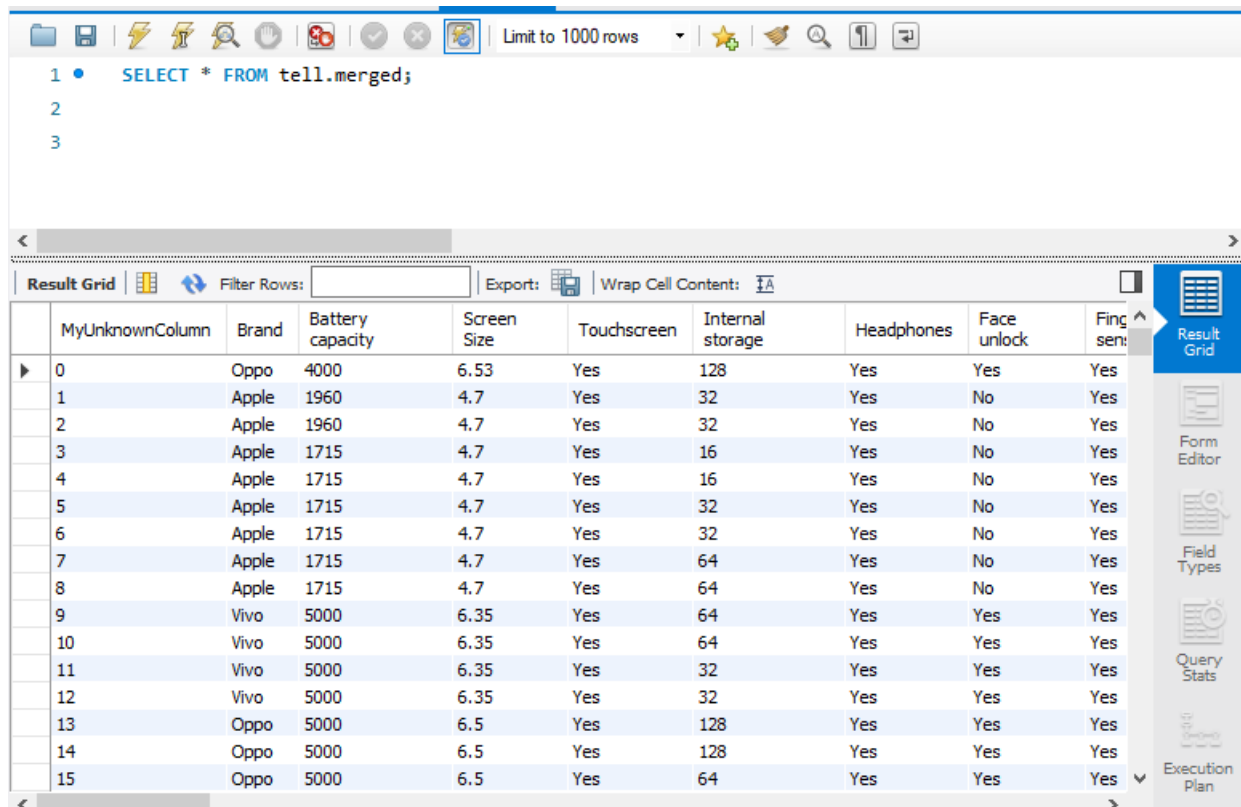
```

Пристапуваме до базата во која сакаме да работиме , и на табелата која веќе е претходно креирано и следно се пренесуваат податоците од csv file-от.



Откако ќе се внесат сите податоци , се креира SQL Table врз која може да се извршуваат разни прашалници.

(tell-името на базата, merged- името на табелата)



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a text area containing the SQL query: `1 • SELECT * FROM tell.merged;`. Below the query, there is a "Result Grid" section. The grid has columns: `MyUnknownColumn`, `Brand`, `Battery capacity`, `Screen Size`, `Touchscreen`, `Internal storage`, `Headphones`, `Face unlock`, and `Fing sen:`. The results are displayed as a table with 16 rows (0 to 15). The right sidebar contains icons for "Result Grid", "Form Editor", "Field Types", "Query Stats", and "Execution Plan".

	MyUnknownColumn	Brand	Battery capacity	Screen Size	Touchscreen	Internal storage	Headphones	Face unlock	Fing sen:
0		Oppo	4000	6.53	Yes	128	Yes	Yes	Yes
1		Apple	1960	4.7	Yes	32	Yes	No	Yes
2		Apple	1960	4.7	Yes	32	Yes	No	Yes
3		Apple	1715	4.7	Yes	16	Yes	No	Yes
4		Apple	1715	4.7	Yes	16	Yes	No	Yes
5		Apple	1715	4.7	Yes	32	Yes	No	Yes
6		Apple	1715	4.7	Yes	32	Yes	No	Yes
7		Apple	1715	4.7	Yes	64	Yes	No	Yes
8		Apple	1715	4.7	Yes	64	Yes	No	Yes
9		Vivo	5000	6.35	Yes	64	Yes	Yes	Yes
10		Vivo	5000	6.35	Yes	64	Yes	Yes	Yes
11		Vivo	5000	6.35	Yes	32	Yes	Yes	Yes
12		Vivo	5000	6.35	Yes	32	Yes	Yes	Yes
13		Oppo	5000	6.5	Yes	128	Yes	Yes	Yes
14		Oppo	5000	6.5	Yes	128	Yes	Yes	Yes
15		Oppo	5000	6.5	Yes	64	Yes	Yes	Yes

## 4. Прашалници и споредба помеѓу Redis и MySQL

### 4.1 Прашалници во Redis

За да се пишуваат прашалници во Redis потребно е поврзување на нашиот сервер со Redis Cloud и да се користи модулот Redi Search кој работи на принцип на индексирање на хешовите. Прашалниците се напишани според синтакса која се користи во тој модул. На пр. `phoned-idx` означува еден телефон (една редица, една торка) од податочното множество.

1. Пронајди ги сите телефони од базата чиј бренд е Apple.

**FT. SEARCH "phone-idx" "@brand:Apple" RETURN all**

2. Пронајди ги сите телефони кои што немаат 5G и се од брендот Apple.

**FT.SEARCH "phone-idx" "@5G:no @brand:Apple" RETURN all**

3. Врати ги конкретните телефони и нивниот бренд кои се поскапи од 20 000.

**FT.SEARCH "phone-idx" "@price>20000" RETURN 3 id brand price**

4. Врати ги телефоните кои се произведени во септември 2019.

**FT.SEARCH "phone-idx" "@launched\_year:2019 @launched\_month:september"**  
**RETURN 4 id launched\_year launched\_month brand**

5. Изброј колку телефони се со резолуција X 750 и резолуција Y 1334, подредени според брендот

**FT.COUNT "phone-idx" "resolution\_x:750 @resolution\_y:1334" ORDER BY brand**  
**RETURN 4 id resolution\_x resolution\_y brand**

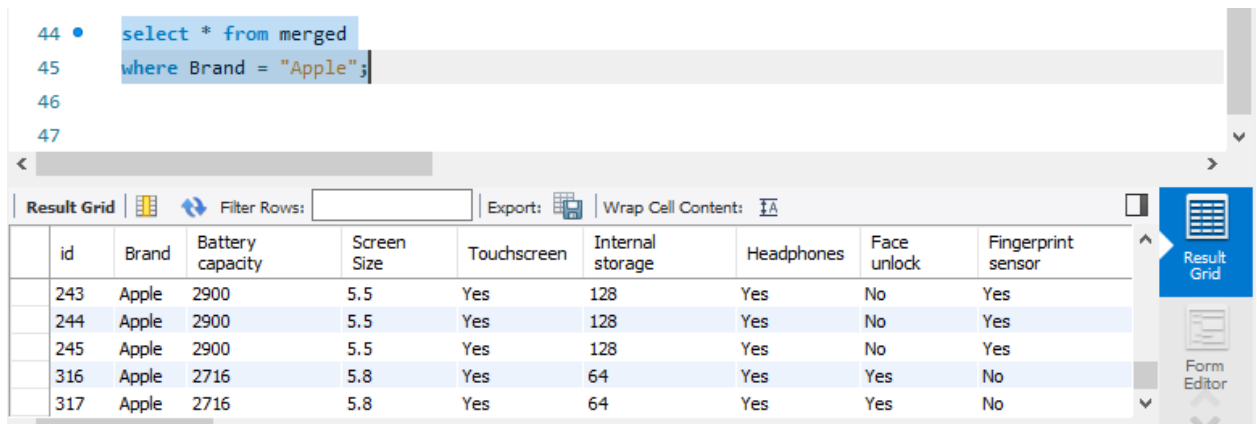
6. Врати го најскапиот телефон од секој бренд, кој што има NFC.

**FT.COUNT "phone-idx" "@NFC:yes" GROUP BY brand RETURN 3 id NFC price**

## 4.2 Прашалници во MySQL

1. Пронајди ги сите телефони од базата чиј бренд е Apple.

**SELECT \* FROM merged**  
**WHERE Brand="Apple" ;**



The screenshot shows a database query interface. The SQL query entered is: `select * from merged where Brand = "Apple";`. Below the query, a table displays the results. The table has 10 columns: id, Brand, Battery capacity, Screen Size, Touchscreen, Internal storage, Headphones, Face unlock, and Fingerprint sensor. The results show 5 rows of data for Apple phones.

id	Brand	Battery capacity	Screen Size	Touchscreen	Internal storage	Headphones	Face unlock	Fingerprint sensor
243	Apple	2900	5.5	Yes	128	Yes	No	Yes
244	Apple	2900	5.5	Yes	128	Yes	No	Yes
245	Apple	2900	5.5	Yes	128	Yes	No	Yes
316	Apple	2716	5.8	Yes	64	Yes	Yes	No
317	Apple	2716	5.8	Yes	64	Yes	Yes	No

Пронајдени се 45 записи за 0.016 sec.



The screenshot shows a database query log with two entries. Each entry includes a query ID, a timestamp, the SQL query, the number of rows returned, and the execution time.

Query ID	Timestamp	Query	Rows Returned	Execution Time
51	17:29:44	select id, Brand from merged where Brand = "Apple" LIMIT 0, 1000	45 row(s) returned	0.000 sec / 0.000 sec
52	17:30:44	select * from merged where Brand = "Apple" LIMIT 0, 1000	45 row(s) returned	0.016 sec / 0.000 sec

2. Пронајди ги сите телефони кои што немаат 5G и се од брендот Apple.

**SELECT \* FROM merged**  
**WHERE 5G="No" AND Brand="Apple";**

```

47 • select * from merged
48 where 5G = "No" and Brand = "Apple";
49
50

```

	Brand	Battery capacity	Screen Size	Touchscreen	Internal storage	Headphones	Face unlock	Fingerprint sensor	Wireless charging	NFC	Number of SIMs	SIM Type	4G/LTE	5G
▶	Apple	1960	4.7	Yes	32	Yes	No	Yes	No	Yes	1	Nano	Yes	No
	Apple	1960	4.7	Yes	32	Yes	No	Yes	No	Yes	1	Nano	Yes	No
	Apple	1715	4.7	Yes	16	Yes	No	Yes	No	Yes	1	Nano	Yes	No
	Apple	1715	4.7	Yes	16	Yes	No	Yes	No	Yes	1	Nano	Yes	No
	Apple	1715	4.7	Yes	32	Yes	No	Yes	No	Yes	1	Nano	Yes	No
	Apple	1715	4.7	Yes	32	Yes	No	Yes	No	Yes	1	Nano	Yes	No

3. Врати ги конкретните телефони и нивниот бренд кои се поскапи од 20 000.

**SELECT** id,Brand,Price **FROM** merged  
**WHERE** Price > 20000;

```

49
50 • select id,Brand,Price from merged
51 where Price > 20000;
52

```

	id	Brand	Price
▶	0	Oppo	23990
	1	Apple	27499
	3	Apple	41999
	4	Apple	23498
	5	Apple	41999
	6	Apple	23498
	7	Apple	41999
	8	Apple	23498
	22	Apple	42900
	23	Apple	51799

Пронајдени се 87 записи.

✓ 63 17:53:32 select id,Brand,Price from merged where Price > 20000 LIMIT 0, 1000 87 row(s) returned

4. Врати ги телефоните кои се произведени во септември 2019.



**SELECT** id,Brand,Launched\_Month,Launched\_Year **FROM** merged  
**WHERE** Launched\_Month="September" AND Launched\_Year="2019";



```

53 • select id,Brand,Launched_Month,Launched_Year from merged
54 where Launched_Month = "September" and Launched_Year ="2019";
55

```

Result Grid				
		Filter Rows:	Export:  Wrap Cell Content: 	
	id	Brand	Launched_Month	Launched_Year
▶	13	Oppo	September	2019
	14	Oppo	September	2019
	15	Oppo	September	2019
	16	Oppo	September	2019
	34	Apple	September	2019
	35	Apple	September	2019
	36	Apple	September	2019
	37	Apple	September	2019
	38	Apple	September	2019
	39	Apple	September	2019

Пронајдени се 54 записи за 0,015 сек.

54 row(s) returned

0.015 sec / 0.000 sec



5. Изброј колку телефони се со резолуција X 750 и резолуција Y 1334, подредени според брендот.

**SELECT** id,Brand,Resolution\_x,Resolution\_y **FROM** merged  
**WHERE** Resolution\_x="750" **AND** Resolution\_y="1334";

```

5
6 • select id,Brand,Resolution_x,Resolution_y from merged
7 where Resolution_x ="750" and Resolution_y="1334";
8

```

Result Grid				
		Filter Rows:	Export:  Wrap Cell Content: 	
	id	Brand	Resolution_x	Resolution_y
	1	Apple	750	1334
	2	Apple	750	1334
	3	Apple	750	1334
	4	Apple	750	1334
	5	Apple	750	1334
	6	Apple	750	1334
	7	Apple	750	1334
	8	Apple	750	1334
	84	Apple	750	1334

Пронајдени се 14 записи.

67 18:04:29 select id,Brand,Resolution\_x,Resolution\_y from merged where Resolution\_x ="750" and Resolution\_y="1334" LIMIT ... 14 row(s) returned

6. Врати го најскапиот телефон од секој бренд, кој што има NFC.

**SELECT** id,Brand,Price,NFC **FROM** merged  
**WHERE** NFC="Yes"

**GROUP BY** Brand  
**HAVING** MAX(Price);

```

58
59 • select id,Brand,Price,NFC from merged
60 where NFC="yes"
61 group by Brand
62 having max(Price);
63

```

Result Grid | Filter Rows: | Export:

	id	Brand	Price	NFC
▶	1	Apple	27499	Yes
	60	OnePlus	39999	Yes
	113	Samsung	54499	Yes
	123	Oppo	25000	Yes
	143	Realme	33999	Yes
	196	Huawei	55649	Yes
	207	Nokia	9999	Yes
	364	Xiaomi	10300	Yes
	371	Google	54999	Yes
	374	Motorola	13849	Yes

## 4.3 Споредба

S.No.REDIS	MYSQL
Развиена е од лабораториите Редис и првично е објавен на 10 мај 2009 година.	Развиена е од Oracle и објавена во мај 1995 година.
Напишана е во ANSI и C јазици.	Напишана е во C и C++.
Основниот модел на база на податоци за Redis е Key-Value Store.	Основниот модел на база на податоци за MySQL е Релационата база на податоци.
Redis е без шема на податоци.	Во MySQL шемата на податоци е фиксна.
Не поддржува XML формат на податоци.	Поддржува XML формат на податоци.
Поддржува секундарни индекси само со модулот RediSearch.	Поддржува секундарни индекси без никакви ограничувања.
Нема Тригери.	Поддржува тригери.
Не го обезбедува концептот на референцијален интегритет. Оттука, нема надворешни клучеви.	Го обезбедува концептот на референцијален интегритет. Оттука, има надворешни клучеви.
Серверските оперативни системи за Redis се BDS, Linux, OS X и Windows.	Серверските оперативни системи за MySQL се FreeBSD, Linux, OS X, Solaris, Windows.
Познати компании како Twinkl Educational Publishing, Merrill Corporation, ASOS.com Limited, Owler, Inc, итн користат Redis.	Famous companies like Airbnb, Pinterest, Slack, Udemy, Twitter, etc uses MySQL.

## 5. Заклучок

После спроведените импорти во двете бази, а и после спроведените прашалници може да се заклучи дека Redis и MySQL се покажаа одлично. Но сепак на големо множество податоци т.н big data, кое нема конкретна структура и кое има можност да се менува неизбежна е потребата од NoSQL база. Redis како база од најбрзиот тип(key-value) одлично се покажа со избраното податочно множество, кое успешно беше импортирано од JSON формат за многу кратко време, односно за помалку од секунда.

Исто така и пребарувањето со прашалниците е многу брзо откако го употребивме сервисот Redi Search.

Како за крај може да се заклучи дека со рапидното зголемување на податоците кои е потребно да се зачувуваат во базите, иднината е во неструктурираните бази на податоци кои нудат флексибилност, скалабилност, огромна брзина.

## 6. Користена литература

- <https://redis.io/topics/mass-insert>
- <https://en.wikipedia.org/wiki/Redis>
- <https://docs.redis.com/latest/rc/databases/import-data/>
- <https://www.datamation.com/storage/8-major-advantages-of-using-mysql/>
- <https://db-engines.com/en/system/MySQL%3BRedis>
- <https://stackshare.io/stackups/mysql-vs-redis>
- <https://tableplus.com/blog/2018/10/redis-vs-mysql-database-comparison.html>