NoSQL ба SQL технологиудын харьцуулалт

Ч.Долгоржав
Монгол Улсын Боловсролын Их Сургууль,
Компьютер Мэдээллийн Технологийн Сургууль,
Сүлжээ Мэдээллийн Системийн Тэнхим
dolgorjav@msue.edu.mn

Хураангуй: Үүлэн дээр байршсан веб программууд өгөгдөл, хэрэглэгчдийн асар их хэмжээтэй гарцаагүй тулгараад байгаа өнөө үед олон жил дангаараа ашиглагдаж байсан холбоост өгөгдлийн сан ашиглахад хүндрэлтэй болсныг сүүлийн жилүүдэд NoSQL технологиор амжилттай шийддэг болоод байна. Энэ өгүүлэл NoSQL гэж юу болох, түүний төрлүүд, SQL ба NoSQL технологиудын гол ялгаа, холбоост өгөгдлийн сангийн хүндрэлийг хэрхэн шийдэж байгаа талаар өгүүлэх болно.

Түлхүүр үгс: *SQL, NoSQL*

1 Удиртгал

Google, facebook, twitter-ийн хэрэглээг өнөөдөр мэдэхгүй хүнгүй шахуу болж, үүнийг дагасан асар их мэдээллийн урсгал секундээр тоологдох хугацаанд дэлхий даяар эргэлдэн байдаг. Мэдээллийн энэ их хэрэгцээ, түүний төрөл зүйл, хүмүүсийн мэдээлэл хүлээж авахыг хүссэн хэлбэр ч олширсоор байна. Яг л энэ асуудлууд бидний ярьж байгаа NoSQL технологийг гарч ирэхэд нөлөөлжээ.

Сүүлийн 15 жилийн дотор вебэд суурилсан программын хэрэглээ огцом нэмэгдэж, томоохон компаниуд хурдацтай өсөн нэмэгдэх хэрэглэгчдийн тоо, төрөл бүрийн өгөгдлийн хэмжээ, бүтэцгүй өгөгдөл зэрэг асуудлуудтай тулгарч эхэлжээ. Харин цаана нь байгаа мэдээлэл бүхэлдээ холбоост өгөгдлийн сан дээр байрлаж байсан нь дээрх асуудлуудтай зохицоход хүндрэл учруулж эхэлсэн байна. Хамгийн гол шалтгаан нь холбоост өгөгдлийн сан нэг машин дээр ажиллахаар бүтэцлэгдсэн байдагт оршино. Өөрөөр хэлбэл тэрхүү том хэмжээний сервер дээр ажиллаж байгаа өгөгдлийн санг өргөтгөхөд тухайн серверийн хэмжээг нэмэгдүүлэх шаардлагатай байв. Дээр нь, өмнө дурдсан асуудлууд дээр

- Веб программын өндөр гүйцэтгэл(байнгын ажиллагаа, дэлхий даяар тархсан хэрэглэгч г.м),
- Нарийн төвөгтэй, уян хатан өгөгдлүүд(social network, multimedia)
- Богино хугацаанд унших, бичих зэрэг зүйлс урган гарч ирэв.

Ингээд 2009 оноос уламжлалт өгөгдлийн сангаас татгалзаж, илүү уян хатан байдлаар ажиллах боломжтой NoSQL технологийг хэрэгжүүлж эхэлсэн байна.

2 NoSQL технологийн тухай

NoSQL гэдэг нь Not Only SQL үгсийн товчлол бөгөөд

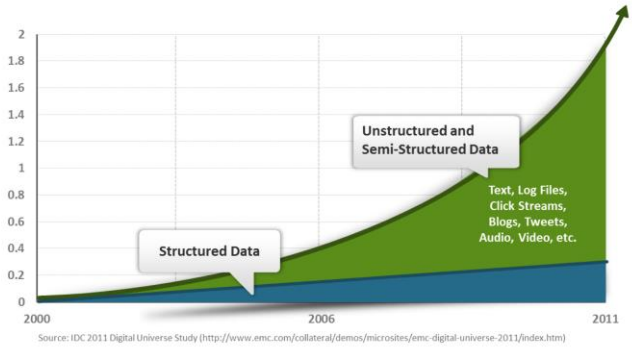
- Холбоосгүй: *ямар ч холбоос байхгүй, холбоост өгөгдлийн сангаас эрс өөр ойлголт*
- Нээлттэй эх бүхий: *Эх кодыг хэн ч чөлөөтэй харж, өөрчилж, компиляци хийж болно*
- Тархсан: *өгөгдөл нь өөр өөр машин дээр хадгалагдахаар зохион байгуулагддаг*
- Хэвтээ чигт өргөтгөсөн: *өгөгдлийн сервер нэмэгдэх бүрт илүү гүйцэтгэл авах боломжтой*

шинж чанартай өгөгдлийн сангуудыг тодорхойлж байгаа юм. Одоогоор Dynamo(Amazon), Cassandra(Facebook), Mongo, CouchDB, BigTable(Google), Neo4J технологиуд энэ талбар дээр голлож байгаа билээ.

Дараах 3 нь өгөгдлийн санг NoSQL чигт хөтөлж буй үндсэн хүчин зүйлс болж байна.

Хэрэглэгчийн тоо: Багахан хугацааны өмнө программуудын хэрэглэгчийн тоо өдөрт 1000, бүр ихсэхдээ 10000 хүрч байлаа. Гэтэл одоо ихэнх программууд үүлэнд байршиж 24/7 хугацаагаар тогтмол ажиллан, 2 тэрбум гаруй онлайн хэрэглэгч, 1 тэрбум гаруй ухаалаг утасны хэрэглээгээр нийтдээ 35 тэрбумаас давсан цагийг онлайн горимд өнгөрөөсөн төдийгүй энэ тоо өдөр өдрөөр өсч байгаа билээ. Энэ том тоонууд хялбар аргаар өргөтгөх боломжтой өгөгдлийн сангийн хэрэгцээг бий болгож байна.

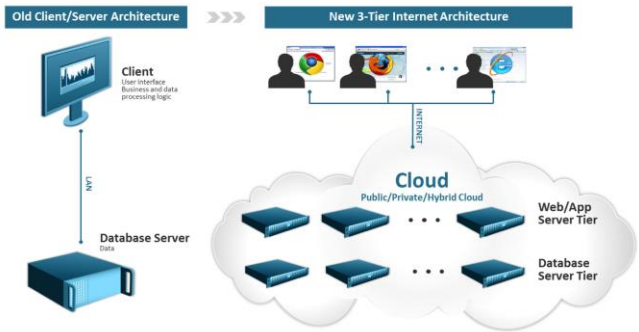
Өгөгдлийн хэмжээ: Нийт онлайн өгөгдлийн хэмжээ эксабайт нэгжээр хэмжигдэхэд хүрсэн бөгөөд нэг эксабайт нь 1 тэрбум гегабайттай тэнцдэг байна. Зөвхөн 2006-2010 оны хооронд энэ тоо 500 хувиар нэмэгдсэн бөгөөд цаашид ч хурдтай өсч байна.



Зураг 1. Олон төрлийн өгөгдлийн хэмжээ өсөн нэмэгдэж байгаа хурдац

Түүнчлэн эдгээр өгөгдлийн 80 гаруй хувь нь текст, блог, дуу болон дүрс г.м бүтэцлэгдээгүй өгөгдлүүд байдаг байна. Гэтэл холбоост өгөгдлийн сан энэ бүхэнтэй ажиллах боломжгүй байсан хэдий ч NoSQL өгөгдлийн сангууд программ болон өгөгдлийн сангийн харилцааг хялбаршуулж, дээр дурдсан төрлийн өгөгдлүүдтэй хялбар ажиллах боломж олгосон билээ.

Үүлэн тооцоолол: Ихэнх программууд саяхан л нэг хэрэглэгчийн, хэрэв олон хэрэглэгчтэй ажилладаг байлаа гэхэд клиент-сервер шийдэлтэй байсан бол өнөөдөр ихэнх шинэ программууд 3 талт интернэт архитектуртай, хувийн болон нийтийн үүлэн дээр ажилладаг, олон тооны хэрэглэгчдийг дэмждэг болоод байна. Программын архитектурын энэхүү шилжилт нь SaaS г.м бизнес загвар, сурталчилгаанд суурилсан загварууд илүү зонхилж байгаа юм.



Зураг 2. Интернэтийн 3 талт архитектур

Энэхүү 3 талт архитектур нь программ руу хөдөлгөөнт төхөөрөмж буюу веб хөтөч ашиглан интернэтээр холбогдох бөгөөд үүлэн дээр ирж байгаа урсгалыг серверийн өргөтгөл рүү чиглүүлэх замаар ачааллыг тэнцвэржүүлдэг. Өөрөөр хэлбэл программын хэрэглэгчдийн тоо 10000 буюу түүнээс дээш гарах тутамд ачааллыг татаж байх дундын сервер үүсгэнэ гэсэн үг юм.

Харин өгөгдлийн сангийн талд нь холбоост өгөгдлийн сан түгээмэл сонголт байсан ч асуудалтай тулгарах болсон шалтгаан нь төвлөрсөн, өргөтгөл нь гадагш биш дээш чиглэдэгтэй холбоотой. Энэ нь түүнийг динамикаар хялбархан өргөтгөхөд төвөг учруулж байлаа. Харин NoSQL технологиуд нь тархсан, гадагш нь өргөтгөсөн байдлаараа 3 талт интернэт архитектурт илүү сайн зохицдог байна.

Эдгээр шалтгаанууд холбоост өгөгдлийн сан дээр суурилан явагдсаар ирсэн программын үйлдвэрлэлийг өөрчилж байгаа бөгөөд хөгжүүлэгчид NoSQL технологи руу эрчимтэй шилжиж байна.

3 NoSQL ба SQL технологиудын үндсэн ялгаа

3.1 Өгөгдлийн загвар

Холбоост болон NoSQL өгөгдлийн загварууд үнэхээр ялгаатай. Холбоост загварт өгөгдлийг маш олон дотоод холбоос бүхий хүснэгт рүү хуваадаг ба хүснэгт бүр мөр баганаас тогтоно. Багана нь ямар нэг тусгай атрибутыг, харин мөрүүд нь бүхий л өгөгдөл мэдээллийг хадгалж байдаг. Хүснэгтүүд гадаад түлхүүр гэж нэрлэгдэх баганаар нөгөө рүүгээ холбогддог. Холбоост загварт өгөгдлийн хэсэг бүр нэг газар байрладгаараа хадгалах зайг багасгадаг ч өгөгдөл хайж олоход энэ нь хүндрэлийг нэмэгдүүлдэг байна. Хүссэн мэдээллийг хэд хэдэн хүснэгтээс цуглуулж гаргаж авах, программд дамжуулахаас өмнө нэгтгэнэ. Гэтэл гаргаж авч байгаа мэдээлэл хэдэн арав, хэдэн зуун хүснэгтээс ч гарч ирэх магадлалтай. Тэгэхээр хамгийн гол зөрчилтэй байгаа газар бол программ өгөгдлөө олох зам, холбоост өгөгдлийн санд өгөгдөл хадгалагдаж байгаа бодит зам 2 юм.

NoSQL нь харин эрс ялгаатай. Дараах үндсэн 4 төрөлд ихэнх NoSQL өгөгдлийн сангууд хуваагддаг.

i. Key-values Stores: (K/V) Энэ нь уламжлалт өгөгдлийн сангийн хэлбэр бөгөөд маш

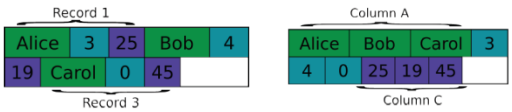
энгийн, түлхүүр ба түүнд харгалзах утгын хослол юм. Түлхүүрийн утга давтагдахгүй бөгөөд бусад утгууд нь strings, integers, floats, byte arrays, болон өгөгдлийн сан интерпретаци хийх хэрэггүй blob төрөлтэй ч байж болно. Энэ хэлбэрийн өгөгдлийн сан их хэмжээний өгөгдөл хадгалах, өргөтгөхөд хялбар бөгөөд тархсан систем дэх өгөгдлийг хурдан, үр дүнтэйгээр удирдах зорилготой юм. K/V өгөгдлийн сан нь бусад NoSQL сангуудын эхлэл ч гэж үздэг.

Key	Value
1	ID:1 Joining Date: 15-July-1985 Designation: Cashier
2	ID:2 Joining Date: 19-March-1982 Designation: Manager
3	ID:3 Joining Date: 4-April-1988 Designation: Front Desk Officer

Зураг 3.K/V загварын өгөгдлийн зохион байгуулалт

ii. Column Oriented Stores: Өгөгдөл хадгалж байгаа нэг блок нь мөрийн биш, баганын өгөгдлийг хадгалах ба SQL-ийн мөр хэлбэрийн өгөгдөлтэй ажиллахаас харьцангуй хурдан байна.

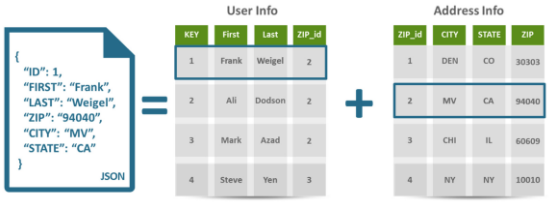
Мөр чигийн хадгалалт *Баганан хадгалалт*



Зураг 4. Мөр болон баганын чигт өгөгдөл хадгалах ялгаа

Баганан хадгалалт бүхий NoSQL өгөгдлийн сан нь кластерийн дагуу тархсан их хэмжээний өгөгдөлтэй ажиллах, түүнийг боловсруулж нэг үр дүн гарган авах үйлдэлд маш хурдан учир OLAP (Online Analytical Processing) болон өгөгдлийн агуулах нь ийм хэлбэрийн өгөгдлийн сантай ажилладаг. Мөн л түлхүүртэй байх хэдий ч тэр нь хэд хэдэн баганыг зааж байдаг байна.

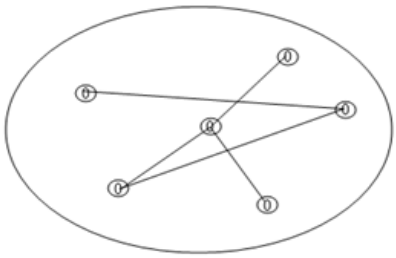
iii. Document Databases: Энэ нь зүгээр уншигдаж болохоор таг-лагдсан документуудыг өгөгдлийн олонлог болгон ашигладаг. Жишээ нь уншиж болохоор xml, json, yaml г.м файлуудаас бүрдэнэ. Тиймээс чөлөөт бүтэцтэй өгөгдлийн санд документын доторх агуулгаас өөр бүтэц хэрэггүй. Документууд бие биеэсээ хамаарахгүй учир хадгалалтад ч асуудал гарахгүй. Бүтэц нь бэхлэгдсэн биш учир документууд ямар ч бүтцийг дэмжинэ. SQL баазад бүх аттрибутууд урьдчилан тодорхойлогдсон байдаг учир утга хадгалаагүй атрибутад ч зориулсан зай байдаг бол документ бүтэцтэй өгөгдлийн санд яг тухайн бичлэгт байгаа атрибутуудыг л аваад, NULL утгатай атрибутыг орхино гэсэн үг.



*Зураг 5. JSON (**J**ava**S**cript **O**bject **N**otation)документад бүх өгөгдлийг хадгалснаар түүнийг уншиж гаргаж авах хурдыг нэмэгдүүлдэг*

JSON форматаар өгөгдлийг шууд документад хадгалдаг. Ийм файл бүр нь программд тусдаа объект хэлбэрээр танигдана. JSON документ нь жишээлэхэд, холбоост өгөгдлийн сангийн 20 хүснэгтийн өгөгдлийг хадгалах боломжтой байна. Мэдээллийн давхардалт байж болох ч хадгалалт нь өндөр өртөгтэй биш, үр дүнд гарч байгаа загвар уян хатан байдлаа алдахгүй, үр ашигтайгаар тархаахад хялбар, унших бичих хурд сайжирсан зэрэг нь үүнийг веб программуудын хувьд нэр хүндтэй болгож байгаа билээ.

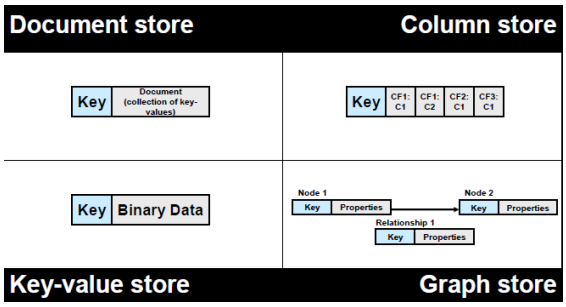
iv. Graph Databases: Энэ нь графын онолд тулгуурлан гарч ирсэн бөгөөд зангилаа, түүний шинж чанар, ирмэгүүд гэсэн үндсэн элемэнтүүд байна.



Зураг 6. Граф өгөгдлийн сангийн дүрслэл

Холбоост өгөгдлийн сантай төстэй нь зангилаануудын хооронд холбоос байх боловч бүтцийн үндсэн элемэнт гэж байхгүй учир ялгаатай зүйл юм. Жишээ нь зангилаа бүр нэг хүнийг заадаг гэвэл А нь В-г танина (дундаа холбоостой), В харин С-г танихгүй (дээрх 2 зангилаа холбоосгүй) гэж дүрсэлж болох юм. Үүнийг зааж байгаа ирмэг бол холбоос гэхээсээ илүү чиглэлийг заана. Одоогоор граф өгөгдлийн сан нь сүлжээн дэх өгөгдлийн боловсруулалтыг үр ашигтай явуулах, зангилаануудын хооронд богино замын тооцоололд үндэслэн байршилд суурилсан үйлчилгээнд ашиглагдаж байна.

Эдгээр төрлүүдэд өгөгдлийг хадгалахад бүгд л түлхүүртэй боловч түлхүүр юуг заах вэ гэдгээрээ ялгагдана.



Зураг 7. NoSQL сангуудын гол төрлүүдэд өгөгдлийг хадгалах дүрслэл

Өөрчлөлтийг удирдах нь холбоост өгөгдлийн сангийн хувьд үнэхээр толгойны өвчин байдаг. Маш бага өөрчлөлтийг хийхэд ч болгоомжтой л байхгүй бол сервер унах, үйлчилгээний түвшинг шууд бууруулах эрсдэлтэй. NoSQL нь дээр дурдсан 4 болон бусад төрлийн аль нь ч дурын төрлийн өгөгдлийг байршуулахад төвөггүй, программын болон өгөгдлийн сангийн өөрчлөлтөд тийм ч их нарийн төвөгтэй ажил шаардахгүй билээ.

3.2 Өргөтгөл, гүйцэтгэл

Байнга өсөн нэмэгдэж байгаа хэрэглэгчийн тоо, өгөгдлийн хэмжээ нь яалт ч үгүй программын өгөгдлийн санг өргөтгөх шаардлагыг авчирдаг.

SQL өгөгдлийн сан нь санах ой, техникийг нэмэх замаар дээш чиглүүлэн өргөтгөж, илүү том серверүүд дээр төвлөрүүлэх замаар өгөгдлийн сангийн хурдан ажиллагааг хангадаг. Үүний утга нь гүйцэтгэлийн нэмэгдэлт ганцхан тухайн сервер дээр нэмэгдэх боломжтой гэсэн үг. Үүнийг up scaling гэнэ. Гэхдээ төвлөрсөн асар их хэмжээний дата гэдэг нь илүү хүчирхэг ч зөвхөн нэг сервер дээр удирдагдаж, зохицуулагдах боломжгүйн учир нь техник бол хязгаарлагдмал зүйл билээ. Гол нь хэрэв өгөгдлийн сан сүлжээн дээр нэг л серверээр удирдагдсан хэвээр байвал түүний гүйцэтгэлд секунд дэх бичилт, уншилтын тоо нэмэгдэхгүй.

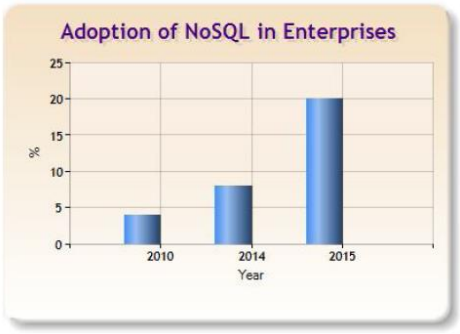
NoSQL өгөгдлийн сан нь тодорхой бүтэц, холбоосгүй бөгөөд ачааллыг дундын серверүүдэд хуваараа тараадаг. Өөрөөр хэлбэл олон тооны(физик болон хийсвэр) дундын серверүүд дээр тархаан ачааллаа шийддэг. Үүнийг гадагш буюу хэвтээ чигийн өртгөтгөл гэж нэрлэх ба энэ нь томоохон өгөгдлийн санг төсөр аргаар шийдэх гарц болдог байна. Олон машиныг кластер хэлбэрээр зохион байгуулж ажиллуулснаар серверүүд бие даан ажиллах бөгөөд хэдийгээр нэг сервер уналаа гэхэд кластер бүхэлдээ ажилласаар байх болно. Үүл бол нэг төрлийн ийм кластер бөгөөд холбоост өгөгдлийн сан үүлтэй төдийлөн сайн зохицдоггүй байна. Үүнийг out scaling гэнэ. Зөвхөн тархсан систем өөрөө өгөгдлийн сангийн гүйцэтгэлийг шууд нэмэгдүүлдэг зүйл биш. Иймд, хэвтээ чигт өргөтгөнө гэдэг нь тооцоолох query-нуудыг ч мөн адил тархааж өөр сервер дээр ажиллуулснаар гүйцэтгэлийг бодитоор нэмэгдүүлэхийг хэлж байгаа юм. Дээр нь уян хатан байдлаа ч алдахгүй. Хэрэглэгчид орох, гарах үед серверийн сан дахь дундын серверүүд автоматаар нэмэгдэж, хасагдаж байдаг. Ингэснээр программын хэрэглээ нэмэгдэхэд түүнийг өөрчлөх шаардлага байхгүй.

3.3 Schema

Өөр нэг үндсэн ялгаа бол холбоост технологи нь schema гэж нэрлэгдэх бүтцийн элемэнттэй байдаг явдал. Холбоост өгөгдлийн санд хадгалагдах өгөгдөлд бүтцийн хатуу тодорхойлолтыг урьдчилан тогтоодог ба өгөгдөл нэгэнт орчихсон л бол түүнийг өөрчлөх нь төвөгтэй болоод эхэлнэ. Гэтэл урьдчилан тооцоолоогүй шинэ өгөгдөл авч эхлэх үед хэрхэх вэ? Том өгөгдлийн эрин үед амьдарч байгаа бидний үед хөгжүүлэгчдийн хувьд энэ бол үнэхээр том асуудал билээ. Гэтэл NoSQL технологид ийм бүтцийн элемэнт байхгүй. Харьцуулбал, документ өгөгдлийн сан нь бүтцийн элемэнтгүй, өөрөөр хэлбэл урьдчилан тодорхойлох ямар ч шаардлагагүйгээр JSON документ руу шууд дурын талбар нэмэх боломж олгож байгаа юм.

4 Дүгнэлт, цаашдын судалгаа

NoSQL өгөгдлийн сангийн K/V, документ, хүснэгт болон граф аль ч төрөл нь технологийн зууны цаашдын ирээдүй болсон веб өргөтгөл, хөдөлгөөнт болон үүлний орчныг дэмждэг байдлыг 2013 онд илүү олон газар туршиж нэвтрүүлнэ гэж Гартнерийн тайлан хэлжээ.



Зураг 8. NoSQL үйлдвэрлэлд нэвтрэх өсөлт

Өгөгдлийн сангийн түүхийг бараг 40 орчим жилийн турш холбоост технологи бичсээр байсан боловч дараах 3 шалтгаанаар хөгжүүлэгчид NoSQL технологи руу эрчимтэй шилжиж байна.

1. Илүү уян хатан өгөгдлийн загвараар дамжуулан программ үйлдвэрлэх илүү сайн нөхцөл,
2. Илүү хэрэглэгч, илүү өгөгдлийг дэмжихэд динамикаар өргөтгөх боломж,
3. Илүү сайн программ, илүү нарийн өгөгдөл боловсруулахыг байнга хүсч байдаг хэрэглэгчийн хүлээлтэд тохирсон гүйцэтгэл.

Хэдийгээр шинэ чиглэл мөн боловч холбоост өгөгдлийн сан энэ талбар дээр гол тоглогч байсаар байгаа бөгөөд ялангуяа жижиг болон жижгэвтэр системүүдийн хувьд энэ нь илүү сайн сонголт болно. Хүснэгтэн зохион байгуулалт бүхий өгөгдөл танил, ойлгомжтойн дээр олон жилийн туршид бат бөх суурь эзэлсэн хүчирхэг хэрэгслүүд энэ технологийг чиглүүлж байдаг билээ.

Иймээс бүтээгдэхүүнээ аль чиглэлд гаргаж байгаагаас хамаарч SQL буюу NoSQL-ийн алийг сонгож авахаа шийдэх хэрэгтэй. NoSQL харьцангуй шинэ учраас нэгэнт SQL орчинд мэргэшсэн хөгжүүлэгчид ийш шилжихэд тодорхой цаг хугацаа шаардах нь дамжиггүй юм. NoSQL дээр хэрхэн query ажиллуулах нь сонирхолтой талбар учраас дараагийн ажил үүнд чиглэгдэнэ.

Ном зүй

[1.] SilvanWeber, “ NoSQLDatabases ” http://www.christof-strauch.de /nosqldbs.pdf
[2.] Vatika Sharma, Meenu Dave, “SQL and NoSQL Databases”, 2012
[3.] An Oracle White Paper, “Oracle NoSQL Database”, September 2011,
[4.] A Couchbase White Paper, “Why NoSQL?”, 2013
[5.] Sidney SHEK, “DISTRIBUTED DATABASES IN THE CLOUD USING NoSQL”, September 2011
[6.] Amir H. Payberah, Not Only SQL (NoSQL) Databases, April 2012
[7.] Keith W. Hare, A Comparison of SQL and NoSQL Databases, May 2011