

ສາທາລະນະ​ລັດ ປະຊາທິປະ​ໄຕ ປະຊາຊົນ​ລາວ

ສັນຕິພາບ ​ເອກະລາດ ປະຊາທິປະ​ໄຕ ​ເອກະ​ພາບ ວັດທະນາ​ຖາວອນ

---------((0))---------



​ ມະຫາວິທະຍາໄລແຫ່ງຊາດ

ຄະນະວິທະຍາສາດທໍາມະຊາດ

ພາກວິຊາວິທະຍາສາດຄອມພິວເຕີ

ບົດລາຍງານ ວິຊາ ວິທີການຄົ້ນຄວ້າ

ສາຂາ (ຕໍ່ເນື່ອງ) ວິທະຍາສາດຄອມພິວເຕີ

ຊື່​ບົດຈົບຊັ້ນ​ (Title)

ພາສາ​ລາວ: ລະບົບຈອງປີ້ລົດເມສາຍໃຕ້ອອນລາຍ

ພາສາ​ອັງກິດ: Southern Bus Ticket Online Booking System

ສະມາຊິກໃນກຸ່ມ (Project Team)

|  |  |  |  |
| --- | --- | --- | --- |
| ລ/ດ | ລະຫັດ | ຊື່ ແລະ ນາມສະກຸນ | ເບີໂທ |
| 1 | 204N0025.19 | ທ ພອນຄຳ ແກ້ວມະນີ | 020 28022677 |
| 2 | ​204N0002.19 | ທ ມະໂນພອນ ມະໂນກຸນ | 020 54000003 |

ສອນໂດຍ ປອ. ລັດສະໝີ ຈິດຕະວົງ

1. ຄວາມສຳຄັນຂອງບັນຫາ

ໃນຍຸກປັດຈຸບັນເຕັກໂນໂລຊີຂໍ້ມູນຂ່າວສານມີຄວາມກ້າວໜ້າ ແລະ ທັນສະໄໝຫຼາຍຂຶ້ນເປັນຕົ້ນແມ່ນເຄື່ຶຂ່າຍ Internet ໄດ້ຂະຫຍາຍຕົວຢ່າງວ່ອງໄວ ຈນກ້າວເຂົ້າສູ່ບົດບາດ ແລະ ຄວາມສຳຄັນຢ່າງຫຼວງຫຼາຍ. ເຮັດໃຫ້ມີການຄິດຄົ້ນສິ່ງອຳນວຍຄວາມສະດວກເພື່ອພັດທະນານຳໃຊ້ໃນຊີວິດປະຈຳວັນຂອງພວກເຮົາ ບໍ່ວ່າຈະເປັນດ້ານການຕິດຕໍ່ສື່ສານ, ການປະຊາສຳພັນຂ່າວ, ການໂຄສະນາ, ການສຶກສາ, ສື່ສານຕ່າງໆ ແລະ ຕະຫຼອດຮອດການຄ້າຂາຍເປັນຕົ້ນກໍ່ໄດ້ນຳເອົາເຕັກໂນໂລຊີເຂົ້າມາຊ່ວຍໃນການດຳເນີນງານໃຫ້ສະດວກສະບາຍວ່ອງໄວ, ນອກຈາກນັ້ນຍັງສາມາດເຜີຍແຜ່ຂໍ້ມູນຂ່າວສານຕ່າງໆຜ່ານເວັບໄຊຕ່າງໆ. ເຮັດໃຫ້ເວັບໄຊໃນເຄື່ອຂ່າຍໃນອິນເຕີເນັດສາມາດຕອບສະໜອງໃນດ້ານຕ່າງໆຜ່ານເວັບໄຊຕ່າງໆ. ເຮັດໃຫ້ເວັບໄຊໃນເຄື່ອຂ່າຍໃນອິນເຕີເນັດສາມາດຕອບສະໜອງໃນດ້ານຕ່າງໆທີ່ກ່າວມານັ້ນເປັນຢ່າງດີ ແລະ ຈະສັງເກດໄດ້ດ້ວຍໜ່ວຍງານອົງກອນທຸລະກິດ, ບໍລິສັດ ຫຼື ຮ້ານຄ້າໃຫ່ຍຕ່າງໆ ຈະມີຄວາມສົນໃຈ ແລະ ແນໃສ່ຄວາມສຳຄັນຂອງສິນຄ້າຫຼາຍຂຶ້ນ ຈຸດປະສົງເພື່ອປະຊາສຳພັນ ຫຼື ການຄ້າຂາຍ, ການສ້າງເວບໄຊຂອງບໍລິສັດ ເພື່ອບໍລິຫານການຂາຍສິນຄ້າກໍຖືວ່າເປັນທາງເລືອກໜຶ່ງທີ່ທັນສະໄໝ ເປັນການອຳນວຍຄວາມສະດວກໃຫ້ແກ່ພະນັກງານ ແລະ ເຈົ້າຂອງບໍລິສັດ, ເປັນການໂຄສະນາບໍລິສັດຜ່ານທາງເວບໄຊໃຫ້ເປັນທີ່ຮູ້ຈັກກັນຫຼາຍຂຶ້ນ ເປັນທາງເລືອກໜຶ່ງທີ່ສະດວກສະບາຍໃນການບໍລິຫານການຂາຍທີ່ລູກຄ້າສາມາດເລືອກຊື້ ແລະ ເບິ່ງສິນຄ້າພາຍໃນບໍລິສັດໄດ້.

ສະຖານີຂົນສົ່ງໂດຍສານສາຍໃຕ້ ເປັນບ່ອນບໍລິການຮັບ-ສົ່ງຜູ້ໂດຍສານ, ສິນຄ້າວັດຖຸສິ່ງ ຂອງ ແລະ ສັດ ຈາກຈຸດໜຶ່ງໄປຫາອີກຈຸດໜຶ່ງ ຊຶ່ງສະຖານີຂົນສົ່ງໂດຍສານທາງໄກສາຍໃຕ້ນີ້ແມ່ນໄດ້ ສ້າງຕັ້ງຂຶ້ນໃນວັນທີ 1 ກັນຍາ 2016 ເຊິ່ງມີລາຍລະອຽດດັ່ງລຸ່ມນີ້: ທີ່ຕັ້ງ ແລະ ພາລະບົດບາດ ຂອງສະຖານນີຂົນສົ່ງທາງໄກສາຍໃຕ້ແມ່ນສະຖານີໜຶ່ງຊຶ່ງຕັ້ງຢູ່ ບ້ສະພັງມຶກ, ເມືອງ ໄຊທານີ, ນະຄອນຫຼວງວຽງຈັນ, ຖະໜົນເລກທີ 450 ປີ ໃກ້ກັບ ສີ່ແຍກໄຟແດງດົງໂດກ. ສະຖານນີຂົນສົ່ງທາງໄກສາຍໃຕ້ ປະກອບມີຫຼາຍໜ່ວຍງານຄື: ອຳນວຍການໃຫ່ຍມີ 1 ທ່ານ, ເລຂານຸການມີ 1 ທ່ານ, ໜ່ວຍງານແຜນການມີ 1 ທ່ານ, ໜ່ວຍງານຮັບ-ຈ່າຍເງິນມີ 1 ທ່ານ, ໜ່ວຍງານບໍລິການຂາຍປີ້ມີ 7 ທ່ານ, ໜ່ວຍງານຮັກສາຄວາມປອດໄພມີ 6 ທ່ານ,ໜ່ວຍງານບໍລິການເຮືອນພັກມີ 8 ທ່ານ ແລະ ບັນດາບໍລິສັດ ທິ່ເຂົ້າມາດໍາ ເນີນທຸລະກິດ ໃນສະຖານຂົນສົ່ງໂດຍສານປະກອບມີ: ບໍລິສັດ ຂົນສົ່ງໂດຍສານຈິດປະສົງ ຍອດນິຍົມ, ບໍລິສັດ ແສງສົມບູນ ຂົນສົ່ງໂດຍສານ, ບໍລິສັດ ແສນສະບາຍ ຂົນສົ່ງໂດຍສານ, ບໍລິສັດ ຈັນທະຈອນ ຂົນສົ່ງໂດຍສານ, ບໍລິສັດ ຈໍາປາສັກ ຂົນສົ່ງໂດຍສານ, ບໍລິສັດ ແສງຈະເລີນ ລົດຕຽງນອນ, ບໍລິສັດ ກຽງໄກ VIP, ບໍລິສດ ສີທອນ ພວງປະເສີດ ລົດຕຽງນອນ. ນອກຈາກນີ້ ສະຖານຍີງ ມີສະຖານທີ່ພັກ , ຮ້ານຄ້າ, ຮ້ານຂາຍຍ່ອຍ, ຮ້ານອາຫານ ແລະ ສິ່ງອໍານວຍຄວາມສະດວກຕ່າງໆ ໄວ້ເພື່ອບໍລິການຜູ້ໂດຍສານທີ່ມາລໍຖ້າລົດໄປຈຸດໝາຍປາຍທາງ.

ເນື່ອງຈາກວ່າ ຈັດການຂໍ້ມູນ, ລາຍງານຂໍ້ມູນຕ່າງໆ, ລວມທັງການຂາຍປີ້ແມ່ນຍັງໃຊ້ແບບຈົດ ແລະ ຜູ້ໂດຍສານທີ່ຕ້ອງການຈອງປີ້ລວງໜ້າ ຕ້ອງໄດ້ໂທຫາພະນັກງານຂາຍປີ້ເພື່ອຈອງ ເຊິ່ງເຮັດໃຫ້ການບໍລິການມີການຊັກຊ້າ ແລະ ຂໍ້ມູນຍັງມີການຕົກເຮ່ຍເສຍຫາຍ

ດັ່ງນັ້ນ , ພວກຂ້າພະເຈົ້າຈຶ່ງເຫັນຄວາມສໍາຄັນຂອງບັນຫາ ຈຶ່ງມີແນວຄວາມຄິດທີ່ຈະສ້າງລະບົບຈອງປີລົດເມ ແບບອອນໄລ ຂອງສະຖານີຂົ່ນສົ່ງໂດຍສານສາຍໃຕນັ້ນຂຶ້ນມາ ເພື່ອຊ່ວຍຫຸດຜ່ອນຄວາມຫຍຸ້ງຍາກໃນການຈອງປີ້ລົດ, ຈັດເກັບຂໍ້ມູນ, ຫຸດຜ່ອນຄວາມຊັກຊ້າໃນການຈັດການຂໍ້ມູນ, ເຮັດໃຫຂໍ້ມູນມີຄວາມເປັນລະບຽບຮຽບຮ້ອຍ ແລະ ເພືອໃຫ້ມີຄວາມສະດວກວ່ອງໄວຕໍ່ການຄົົນຫາຂໍ້ມູນ.

1. ຈຸດປະສົງຂອງການຄົ້ນຄ້ວາ (Objectives)

* ເພື່ອສຶກສາບັນຫາທີ່ເກີດຂຶ້ນໃນປະຈຸບັນ ແລະ ຄວາມຕ້ອງການຂອງລະບົບ.
* ເພື່ອສ້າງລະບົບຂາຍປີ້ລົດອອນໄລຂອງສະຖານີຂົນສົ່ງໂດຍສານສາຍໃຕ້.
* ເພື່ອສ້າງຮູບແບບການຈັດການຂໍ້ມູນການໃຫບໍລິການ.
* ເພື່ອເຜີຍແຜ່ຂໍ້ມູນການຂາຍປີ້ລົດເມຂອງສະຖານຂົນສົ່ງໂດຍສານສາຍໃຕ້.
* ເພື່ອການລາຍງານໃຫ້ສະດວກ ແລະ ຖືກຕ້ອງ.

1. ຂອບເຂດການຄົ້ນຄ້ວາ (Scope)

ລະບົບຈອງປີ້ລົດເມສາຍໃຕ້ແບບອອນລາຍ ເປັນລະບົບແບບ Client-Server Web Application ເຊິ່ງປະກອບດ້ວຍໜ້າວຽກຫຼັກດັ່ງນີ້:

* ຈັດການຂໍ້ມູນພື້ນຖານ : (ຂໍ້ມູນພະນັກງານ, ຂໍ້ມູນລົດ, ຂໍ້ມູນປະເພດລົດ, ຂໍ້ມູນສາຍທາງ)
* ສະໝັກສະມາຊິກ
* ບໍລິການ : (ຈອງປີ້, ອອກປີ້)
* ລາຍງານ : (ລາຍງານຂໍ້ມູນການຈອງ, ລາຍງານຂໍ້ມູນພະນັກງານ, ລາຍງານຂໍ້ມູນສາຍທາງ, ລາຍງານຂໍ້ມູນລົດ, ລາຍງານຂໍ້ມູນອອກປີ້)

1. ປະໂຫຍດຄາດວ່າຈະໄດ້ (Expected Outcome of the Project)

* ໄດ້ລະບົບຈອງປີ໊ລົດເມແບບອອນລາຍຂອງສະຖານີຂົ່ນສົ່ງສາຍໃຕ້
* ໄດ້ລະບົບທີ່ຈະຊ່ວຍແກ້ໄຂບັນຫາການຈອງຈອງໄດ້ສະດວກ ແລະ ວ່ອງໄວຂຶ້ນ
* ໄດ້ລະບົບຊ່ວຍເພີ່ມຊ່ອງທາງໃນການຂາຍປີ້ໃຫ້ກັບຜູ້ປະກອບການ
* ມີລະບົບເຜີຍແຜ່
* ໄດ້ລະບົບທີ່ສາມາດສ້າງລາຍງານໄດ້ຢ່າງສະດວກ ແລະ ຖືກຕ້ອງ

1. ວິທີດຳເນີນການຄົ້ນຄ້ວາ (Research Methodology)

ຂັ້ນຕອນວິທີການດຳເນີນການພັດທະນາລະບົບໃນຄັ້ງນີ້ແມ່ນອີງໃສ່ລັກສະແນແບບຈຳລອງຂອງ Adapted Waterfall Model ເປັນແນວທາງໃນການວິເຄາະ ແລະ ອອກແບບລະບົບ ເຊິ່ງມີໜ້າວຽກດັ່ງນີ້:

ວາງແຜນ

ວິເຄາະ

ອອກແບບ

ພັດທະນາ

ທົດສອບ

ສ້າງເອກະສານ

ຮູບທີ່ 1 ວົງຈອນການພັດທະນາແບບນ້ຳຕົກ (Adapted Waterfall Model)

* + ວາງແຜນ

ດຳເນີນການຈັດຕັ້ງກຸ່ມຂຽນບົດໂຄງການພຽງພ້ອມກັບການກຳນົດຂໍ້ຂອງໂຄງການກຳນົດຫົວຂໍ້ຂອງໂຄງການ. ຈາກນັ້ນ, ກໍໄດ້ລົງເກັບກຳຂໍ້ມູນຢູ່ສະຖານນີຂົຍສົ່ງໂດຍສານສາຍໃຕ້ ເພື່ອໃຫ້ໄດ້ຂໍ້ມູນ ແລະ ຂັ້ນຕອນການເຮັດວຽກໂດຍລວມກ່ຽວກັບການເຮັດວຽກຕ່າງໆຂອງສະຖານີດັ່ງກ່າວ.

* + ວິເຄາະ

ໄລຍະນີ້ພວກເຮົາຈະສຶກສາລະບົບເກົ່າ ແລະ ຄວາມຕ້ອງການຂອງຜູ້ໃຊ້. ຈາກນັ້ນ, ກໍນຳເອົາຂໍ້ມູນຕ່າງໆລວບລວມເອົາຂໍ້ມູນຕ່າງໆທີ່ລວບລວມໄດ້ມາເພື່ອວິເຄາະເປັນຂໍ້ກຳນົດຄວາມຕ້ອງການຂອງລະບົບໃໝ່ພ້ອມທັງແຕ້ມແບບຈຳລອງ DFD ເພື່ອສະແດງເຖິງການໄຫຼຂໍ້ມູນໄປເຖິງຂະບວນການ ແລະ ແຜນວາດ E-R Diagram ເພື່ອສະແດງຄວາມສຳພັນລະຫວ່າງຂໍ້ມູນ.

* + ອອກແບບ

ໄລຍະການອອກແບບ ແມ່ນໄລຍະທີ່ພວກຂ້າພະເຈົ້າ ນໍາເອົາຜົນທີ່ໄດ້ຈາກການວິເຄາະມາອອກແບບລະບົບໂດຍການອອກແບບ, ໜ້າຟອມປ້ອນຂໍ້ມູນ ແລະ ຫນ້າຟອມລາຍງານໂດຍນໍາໃຊ້ Visual Studio code ລວມທັງອອກແບບຖານຂໍ້ມູນ.

* ພັດທະນາ

ດຳເນີນການສ້າງຂໍ້ມູນຕາມທີ່ໄດ້ເຮັດ Data Dictionary ໃນຂັ້ນຕອນກ່ອນໜ້ານັ້ນ. ຈາກນັ້ນ, ກໍລົງມືປະຕິບັດຂຽນ Code ເພື່ອຕິດຕໍ່ ແລະ ເຮັດວຽກກັບຖານຂໍ້ມູນທີ່ໄດ້ສ້າງໄວ້ແລ້ວ.

* + ທົດສອບ

ສ້າງແບບວິທີການທົດສອບລະບົບ ແລະ ນຳເອົາຊຸດຂໍ້ມູນທີ່ນຳມາທົດສອບຢ່າງໜ້ອຍ 20 records ມາດຳເນີນການທົດສອບແລ້ວສະຫຼຸບຜົນໄດ້ຮັບ ແລະ ທຳການປັບປຸງແກ້ໄຂຈົນໄດ້ຜົນອອກທີ່ຖືກຕ້ອງ

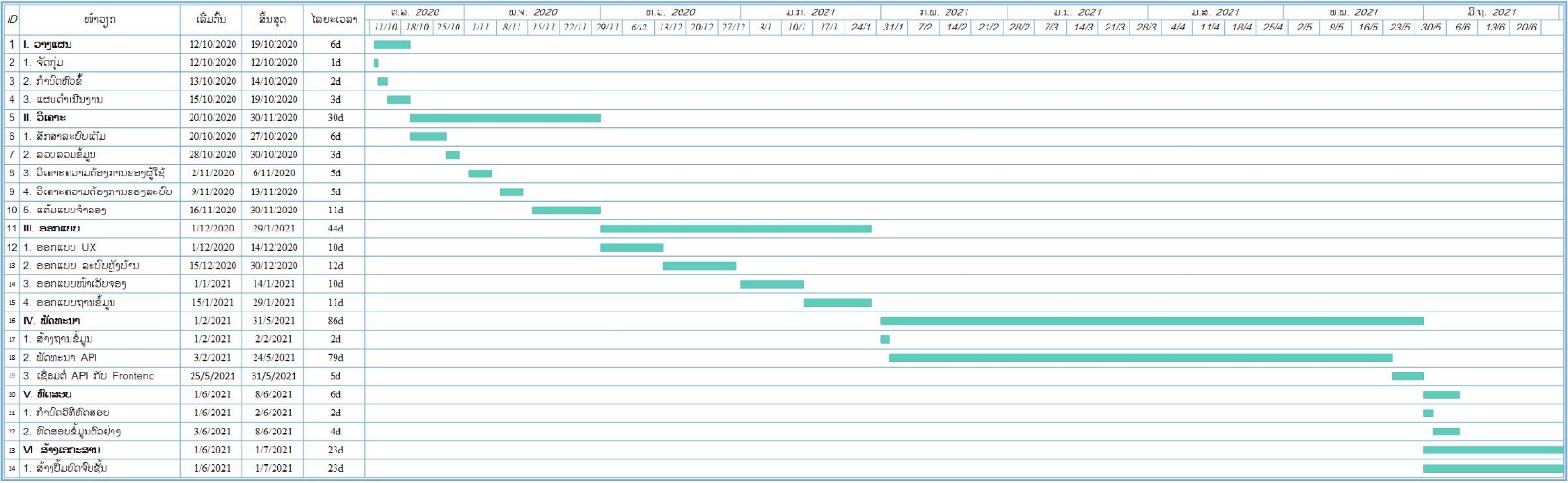
* ສ້າງເອກະສານ

ສ້າງປຶ້ມບົດຈົບຊັ້ນ ລະບົບການຈອງປີ້ແບບອອນລາຍຂອງສະຖານີຂົນສົ່ງໂດຍສານສາຍໃຕ້ ແລ້ວສົ່ງມອບປຶ້ມພ້ອມທັງແຜ່ນ CD ທີ່ປະກອບດ້ວຍ Source Code ,ເອກະສານບົດຈົບຊັ້ນ ແລະ Presentations.

1. ສະຖານທີ່ສຶກສາ (Study Site)

ສະຖານີຂົນສົ່ງ ໂດຍສານວຽງຈັນ ຈຳກັດ (ສາຍໃຕ້-ຕ່າງປະເທດ)

1. ​ໄລຍະປະຕິບັດ (Duration)



1. ເຄື່ອງມື​​ທີ່ນຳໃຊ້​ໃນ​ການ​ພັດທະນາ (Development Tools)

|  |  |
| --- | --- |
| 1. Hardware | 1. Software |
| * ຄອມພິວເຕີ Lenovo Intel(R) Core(TM) I5-3230M CPU @2.60GHz, RAM 8GB DDR3L 1600MHz, SSD 240GB,HDD 500GB * USB 16GB 1ອັນ | * ລະບົບປະຕິບັດການ Windows 10 Professional 64 Bit * Microsoft Visio 2016 ໃຊ້ແຕ້ມແຜນວາດການໄຫຼຂໍ້ມູນ (DFD, ER, Flowchart) * Adobe XD ໃຊ້ອອກແບບ UX/UI * Studio 3T For MongoDB ແລະ Moon Modeler ໃຊ້ຈັດການຖານຂໍ້ມູນ ແລະ ອອກແບບ Database Model * Visual Studio Code ໃຊ້ຂຽນໂຄດດ້ວຍພາສາ JavaScript (ReactJS, NodeJS, GraphQL) * MS Office 2013 Professional ໃຊ້ເພື່ອສ້າງເອກະສານຕ່າງໆ, ບົດນຳສະເໜີ ແລະ ເຮັດປຶ້ມບົດຈົບຊັ້ນ |

1. ເອກະສານອ້າງອີງ (Reference)

ສົມມິດ ທຸມມາລີ, ແລະ ອາມອນ ຈັນທະພາວົງ. (2012). ການວິເຄາະ ແລະ ອອກແບບລະບົບ (System Analysis and Design), ໜ້າ 56, 68 ນະຄອນຫຼວງວຽງຈັນ: ໂຮງພິມນະຄອນຫຼວງວຽງຈັນ

1. ທົບທວນທິດສະດີ ແລະ ບົດທີ່ກ່ຽວຂ້ອງ
   1. ທົບທວນທິດສະດີທີ່ກ່ຽວຂ້ອງ

ລະບົບຈອງປີ້ລົດເມສາຍໃຕ້ອອນລາຍແມ່ນພັດທະນາຂຶ້ນໂດຍນຳໃຊ້ທິດສະດີ ຫຼື ຄວາມຮູ້ຈາກ 3 ສ່ວນຄື: ທິດສະດີໃນການວິເຄາະ ແລະ ອອກແບບລະບົບ, ທິດສະດີກ່ຽວກັບການສ້າງຖານຂໍ້ມູນ ແລະ ຄວາມຮູ້ກ່ຽວກັບພາສາໃນການພັດທະນາລະບົບ. ລາຍລະອຽດຂອງແຕ່ລະທິດສະດີແມ່ນຈະໄດ້ນຳສະເໜີໂດຍສັງເຂບດັ່ງຕໍ່ໄປນີ້:

* + 1. ທິດສະດີກ່ຽວກັບການວິເຄາະ ແລະ ການອອກແບບລະບົບ

ການ​ວິ​ເຄາະ​ລະ​ບົບ​ເປັນ​ຂະ​ບວນ​ການ​ທຳ​ຄວາມ​ເຂົ້າ​ໃຈ ແລະ ກຳ​ນົດ​ລາຍ​ລະ​ອຽດ​ຂອງ​ບັນ​ຫາ​ເພື່ອ​ຈະ​ໄດ້​ພິ​ຈາ​ລະ​ນາ​ນຳ​ເອົາ​ລະ​ບົບ​ຂໍ້​ມູນ​ຂ່າວ​ສານ​ໃດ​ເຂົ້າ​ໄປ​ແກ້​ບັນ​ຫາ​ເລົ່າ​ນັ້ນ, ສ່ວນ​ການ​ອອກ​ແບບ​ລະ​ບົບ​ໝາຍ​ເຖິງ​ຂະ​ບວນ​ການ​ກຳ​ນົດ​ລາຍ​ລະ​ອຽດ​ຕ່າງໆ​ວ່າ​ຈະ​ຕ້ອງ​ເຮັດ​ແນວ​ໃດ​ກັບ​ອົງ​ປະ​ກອບ​ຂອງ​ລະ​ບົບ​ຂໍ້​ມູນ​ຂ່າວ​ສານ​ເພື່ອ​ຈະ​ໄດ້​ນຳ​ໄປ​ໃຊ້​ໃຫ້​ເກີດ​ຜົນ​ໃນ​ທາງ​ພາຍນອກ​ໄດ້. ສິ່ງ​ທີ່​ນຳ​ສະ​ເໜີ​ໃນ​ຫົວ​ຂໍ້​ນີ້​ໄດ້​ຂັດ​ສັນ​ພາກ​ສ່ວນ​ໜຶ່ງ​ຈາກ​ (ພ​ຣະ ເຫຼົາ​ຄຳ ເພັດ​ວິ​ໄລ, ວິ​ລະ​ຍຸດ ວົງ​ທິ​ລາດ ແລະ ເພັດ​ດາວວອນ ທິ​ແກ້ວ, 2016) ແລະ (ສົມ​ມິດ ທຸມ​ມາ​ລີ ແລະ ອາ​ມອນ ຈັນ​ທະ​ພາ​ວົງ​, 2012).

* + - 1. ວົງຈອນພັດທະນາລະບົບ
      2. ແຜນວາດການໄຫຼຂໍ້ມູນ (Data Flow Diagram DFD)

- ເປັນແຜນພາບທີ່ສະຫຼຸບລວມຂໍ້ມູນທັງໝົດໄດ້ຈາກການວິເຄາະໃນລັກສະນະຂອງຮູບແບບທີ່ເປັນໂຄງສ້າງ.

- ເປັນຂໍ້ຕົກລົງຮ່ວມກັນລະຫວ່າງນັກວິເຄາະລະບົບ ແລະ ຜູ້ຊົມໃຊ້.

- ເປັນແຜນພາບທີ່ໃຊ້ໃນການພັດທະນາຕໍ່ໃນຂັ້ນຕອນຂອງການອອກແບບ.

- ຮູ້ທີ່ໄປທີ່ມາຂອງຂໍ້ມູນທີ່ໄຫຼໃນຂະບວນການຕ່າງໆ.

* + - 1. ສັນຍະລັກທີ່ໃຊ້ໃນແຜນວາດຂໍ້ມູນ

ຕາຕະລາງ ສະແດງສັນຍາລັກ **Data Flow Diagram**

|  |  |  |
| --- | --- | --- |
| ຊື່ | ສັນຍາລັກ | ຄວາມຫມາຍ |
| Boundary Or External Entity |  | ຂອບເຂດຫມາຍເຖິງພາກສ່ວນທີ່ກ່ຽວຂ້ອງເກັບລະບົບເຊິ່ງລະບົບບໍ່ສາມາດຄວບຄຸມໄດ້ |
| Process |  | ປະມວນຜົນຫຼືຫນ້າວຽກທີ່ເຮັດໃນໂຄງການນັ້ນໆ |
| Data Store |  | ບ່ອນຈັດເກັບຂໍ້ມູນ |
| Data Flow |  | ການໄຫຼຂອງຂໍ້ມູນ |
| Real-Time Link |  | ການເຊື່ອມໂຍງແບບໄກທີ່ມີການຕອບກັບແບບທັນທີທັນໃດ |

* + - 1. ຫຼັກການຂຽນແຜນວາດການໄຫຼຂໍ້ມູນ

1. Process:
   * ເມື່ອມີຂໍ້ມູນເຂົ້າໄປທີ່ Process ກໍ່ຕ້ອງມີຂໍ້ມູນ ຫຼື ຜົນຮັບອອກມາຈາກ Process ເຊັ່ນກັນຈະເປັນໄປບໍ່ໄດ້ທີ່ມີສະເພາະຂໍ້ມູນເຂົາຢ່າງດຽວ.
2. Data store:

* ຂໍ້ມູນຈະໄຫຼຈາກ Data store ໜຶ່ງໄປຫາ Data store ໜຶ່ງໂດຍກົງບໍ່ໄດ້ຈະຕ້ອງຜ່ານ Process ເທົ່ານັ້ນ
* ຂໍ້ມູນທີ່ສົ່ງຜ່ານ External entityບໍ່ສາມາດໄຫຼເຂົ້າໄປ Data storeໂດຍກົງໄດ້ຈະຕ້ອງໃຊ້ Processເປັນຕົວກາງໃນການເຊື່ອມໂຍງເພື່ອຈັດເກັບຂໍ້ມູນໃນ Data store
* ຂໍ້ມູນທີ່ໄຫຼຜ່ານຈາກ Data storeບໍ່ສາມາດເຊື່ອໂຍງເຂົ້າກັບ External entity ໄດ້ໂດຍກົງຈະຕ້ອງຜ່ານ Processເທົ່ານັ້ນ.

1. External entity:

* External entity ບໍ່ສາມາດເຊື່ອມໂຍງເຂົ້າຫາກັນໄດ້ຈະຕ້ອງໃຊ້ Process ເປັນຕົວກາງເພື່ອສົ່ງຜ່ານ ແລະ ຊື່ຂອງ External entity ຈະໃຊ້ຄໍານາມເທົ່ານັ້ນ.

1. Data flow:

* ການໄຫຼຂໍ້ມູນທີ່ມີຫົວຊີ້ໄປທີ່ Processໝາຍເຖິງ Processມີການອ່ານ ຫຼື ການດືງຂໍ້ມູນຈາກ Data store ມາໃຊ້ວຽກ
* ການໄຫຼຂໍ້ມູນຈາກ Process ທີ່ມີຫົວລູກສອນຊີ້ໄປຍັງ Data store ໝາຍເຖິງການ Update ຫຼື ການເພີ່ມຂໍ້ມູນລົງໄປທີ່ Data store

|  |  |
| --- | --- |
| ອະນຸຍາດ | ບໍ່ອະນຸຍາດ |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* + - 1. Flowchart
    1. ທິດສະດີກ່ຽວກັບຖານຂໍ້ມູນ

Database ຫຼື ຖານຂໍ້ມູນຄືກຸ່ມຂອງຂໍ້ມູນທີ່ຖືກເກັບລວບລວມໄວ້ໂດຍມີຄວາມສໍາພັນເຊິ່ງກັນ ແລະ ກັນໂດຍບໍ່ໄດ້ບັງຄັບວ່າຂໍ້ມູນທັງຫມົດນີ້ຈະຕ້ອງເກັບໄວ້ໃນແຟ້ມຂໍ້ມູນດຽວກັນ ຫຼື ແຍກເກັບຫຼາຍໆແຟ້ມຂໍ້ມູນ

ລະບົບຖານຂໍ້ມູນຄືລະບົບທີ່ລວບລວມຂໍ້ມູນຕ່າງໆທີ່ກ່ຽວຂ້ອງກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງມີລະບົບ, ມີຄວາມສໍາພັນລະຫວ່າງຂໍ້ມູນຕ່າງໆທີ່ຊັດເຈັນໃນລະບົບຖານຂໍ້ມູນຈະປະກອບດ້ວຍແຟ້ມຂໍ້ມູນຫຼາຍແພ້ມທີ່ມີຂໍ້ມູນກ່ຽວຂ້ອງກັນ,ສໍາພັນກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງເປັນລະບົບ ແລະ ເປີດໂອກາດໃຫ້ ຜູ້ໃຊ້ສາມາດໃຊ້ງານ ແລະ ຮັກສາປ້ອງກັນຂໍ້ມູນເຫຼົ່ານີ້ໄດ້ຢ່າງມີປະສິດທິພາບໂດຍມີຊອບແວຣ໌ທີ່ປຽບສະເຫມືອນຊື່ກາງລະຫວ່າງຜູ້ໃຊ້ ແລະໂປຣແກຣມຕ່າງໆທີ່ກ່ຽວຂ້ອງກັບການໃຊ້ຖານຂໍ້ມູນເອີ້ນວ່າລະບົບຈັດການຖານຂໍ້ມູນ ຫຼື DBMS

* + - 1. ການເຮັດ Normalization

1. ຄວາມໝາຍ Normalization

Normalization ເປັນຫຼັກການໜຶ່ງງທີີຜູູອອກແບບຖານຂໍ້ມູນຈະຕ້ອງນໍາມາໃຊ້ໃນການແປງຂໍ້ມູນທີີຢູູໃນຮູບແບບທີີຊໍໍາຊ້ອນໃຫ້ຢູູໃນຮູບແບບທີີງ່າຍຕໍ່ການນໍາໄປໃຊ້ງານ ແລະ ກໍ່ໃຫູເກີດບັນຫານູອຍທີີສຸດ.

1. ຈຸດປະສົງຂອງການເຮັດການເຮັດ Normalization

* ຫຼຸດຜ່ອນຄວມຊໍ້າຊ້ອນຂອງຂໍ້ມູນ ເມື່ອຫຼຸດຄວາມຊໍ້າຊ້ອນຈະເຮັດໃຫ້ຫຼຸດເນື້້ອທີ່ໃນການຈັດເກັບຂໍ້ມູນ
* ຫຼຸດບັນຫາຄວາມບໍ່ຖຶກຕ້ອງຂອງຂໍ້ມູນ ເມື່ອຂໍ້ມູນບໍ່ເກີດຄວາມຊໍ້າຊ້ອນ ເຮັດໃຫ້ການປັບປຸງຂໍ້ມູນສາມາດເຮັດໄດ້ຈາກແຫຼ່ງຂໍ້ມູນພຽງບ່ອນດຽວ.
* ຫຼຸດຄວາມຜິດພາດທີ່ເກີດຈາກການປັບປຸງຂໍ້ມູນ

1. ຂັ້ນຕອນການເຮັດ Normalization

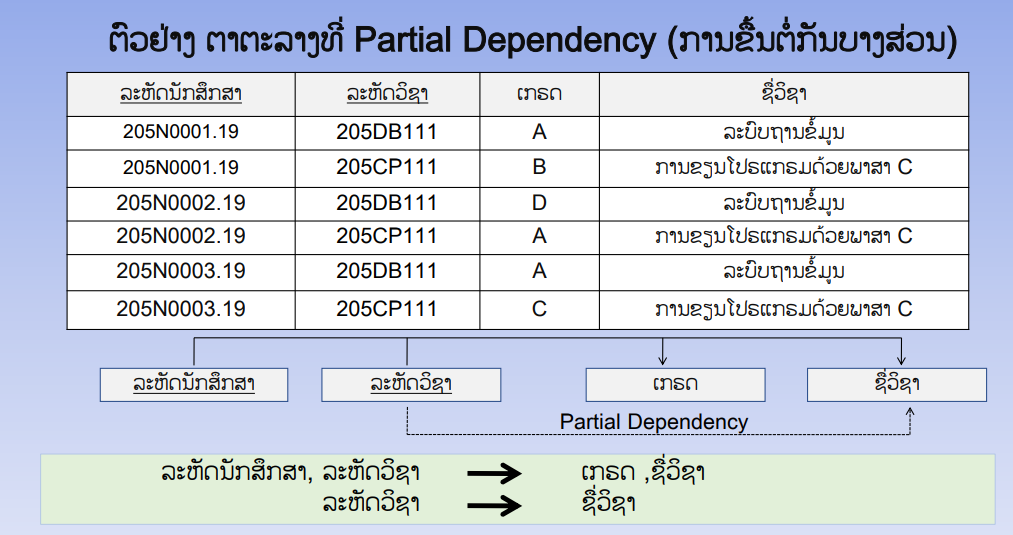
* Normalization ລະດັບ 1 ຫຼືເອີ້ນວ່າ 1NF
* Normalization ລະດັບ 2 ຫຼືເອີ້ນວ່າ 2NF
* Normalization ລະດັບ 3 ຫຼືເອີ້ນວ່າ 3NF

1. First Normal Form (1NF)

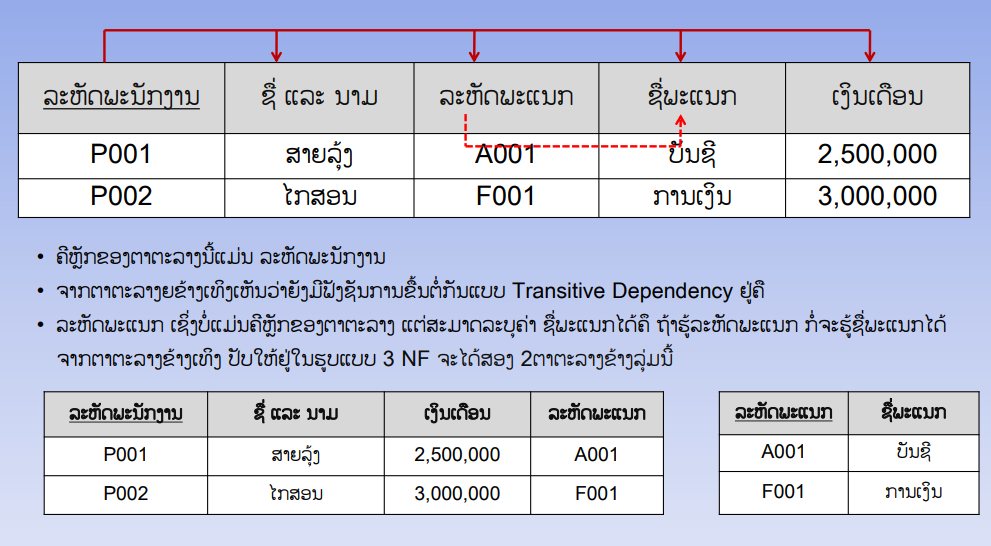
* ທຸກ Attribute ໃນແຕ່ລະ record ຈະເປັນ single value ບໍ່ມີ ຄ່າຂອງກຸ່ມຂໍ້ມູນຊໍ້າຊ້ອນກັນ (Repeating Group).
* ຂໍ້ມູນທຸກແຖວ (Tuple) ຕ້ອງມີຄ່າບໍ່ຊໍ້າກັນ

1. Second Normal Form (2NF)

* ຕ້ອງເປັນ First Normal Form (1NF) ມາກ່ອນ
* ຕ້ອງບໍ່ມີ Partial Dependency (ການຂຶ້ນຕໍ່ກັນບາງສ່ວນ)
* ສະຫຼຸບ: ການເຮັດນໍມາລາຍເຊຊັນ ລະດັບ2 (Second normal form : 2NF) ເປັນການເຮັດໃຫ້ແອດທິບິວທີີ່ບໍ່ຂຶ້ນກັບຄີຫຼັກອອກໄປເພື່ອໃຫ້ແອດທິບິວອື່ນທັງໝົດຂຶ້ນກັບສ່ວນທີີ່ເປັນຄີຫຼັກ



1. Third Normal Form (3NF)

* Relation ນັ້ນຈະຕ້ອງມີຄຸນສົມບັດ 2NF
* ຕ້ອງບໍ່ມີຄວາມສໍາພັນລະຫວ່າງ Non-key Attribute ຫຼື ບໍ່ມີ Transitive Dependency
* ສະຫຼຸບ: ແອດທິບິວທີີ່ບໍ່ແມ່ນຄີຫຼັກ ຕ້ອງບໍ່ຂຶ້ນຕໍ່ກັນເອງ

1. ສະຫຼຸບ Normalization

• 1NF ທຸກແອດທິບິວໃນແຕ່ລະແຖວຕ້ອງມີຂໍ້ມູນພຽງຄ່າດຽວເທົ່ານັ້ນ

• 2NF ຣີເລເຊິນນັ້ນຕ້ອງບໍ່ມີຄວາມສໍາພັນລະຫວ່າງແອດທິບິວແບບບາງສ່ວນ(ແອດທິບິວ ທຸກຕົວຕ້ອງຂຶ້ນກັບຄີຫຼັກທຸກຕົວບໍ່ຂຶ້ນຢູູ່ກັບຕົວໃດຕົວໜຶ່ງ)

• 3NF ທຸກແອດທິບິວທີີ່ບໍ່ແມ່ນຄີຫຼັກບໍ່ມີຄຸນສົມບັດໃນການກໍານົດຄ່າຂອງແອດທິບິວຕົວອື່ນ

* + - 1. ແຜນວາດຄວາມສຳພັນ Entity (ER Diagram)

ການອອກແບບຈໍາລອງຖານຂໍ້ມູນ ER ນັ້ນກ່ອນອື່ນຕ້ອງຮູ້ຈັກຄວາມໝາຍສັນຍາລັກຕ່າງໆທີ່ໃຊ້ໃນການອອກແບບຈໍາລອງ ER ໄດ້ຢ່າງຖືກຕ້ອງດັ່ງນັ້ນໃນຫົວຂໍ້ນີ້ຈະເວົ້າເຖິງຄວາມໝາຍ ແລະ ການໃຊ້ງານສັນຍາລັກຕ່າງໆຂອງ ER.

* 1. ເອັນຕີຕີ້ (Entity)

ເອັນຕີຕີ້ຄືວັດຖຸທີ່ເຮົາສົນໃຈເຊິ່ງອາດເປັນໄດ້ທັງບຸກຄົນ, ສະຖານທີ່, ວັດຖຸ, ເຫດການ ຫຼື ແນວຄິດທີ່ກໍ່ໃຫ້ເກີດກຸ່ມຂອງຂໍ້ມູນທີ່ຕ້ອງການເອັນຕີຕີ້ແບ່ງອອກເປັນ 2 ປະເພດຄື:

* 1. Strong Entity:

ເປັນເອັນຕີຕີ້ທີ່ເກີດຂຶ້ນດ້ວຍຕົນເອງເປັນອິດສະຫຼະບໍ່ຂຶ້ນກັບເອັນຕີຕີໃດສັນຍາລັກທີ່ໃຊ້ຄືຮູບສີ່ຫຼ່ຽມ ແລະ ສາມາດເອີ້ນ Strong Entity ໄດ້ອີກຊື່ໜື່ງວ່າ Regular Entity.

CUSTOMER

STOCK

ຮູບທີ 3 ຮູບStrong Entity

* 1. Weak Entity:

ເອັນຕີຕີ້ຊະນິດນີ້ຈະຂຶ້ນກັບເອັນຕີຕີ້ຊະນິດອື່ນໆບໍ່ສາມາດເກີດຂຶ້ນໄດ້ຕາມລໍາພັງ ແລະ ຈະຖືກລົບເມື່ອເອັນຕີຕີ້ຫຼັກຖືກລົບອອກ ສັນຍາລັກທີ່ໃຊ້ຄືຮູບສີ່ຫຼ່ຽມຊ້ອນກັນ.

CUSTOMER

STOCK

ຮູບທີ 4 ຮູບWeak Entity

* 1. ແອັດທຣິບີວ (Attribute)

ແອັດທີຣບີວຄືຄຸນສົມບັດຂອງສົມບັດຂອງເອັນຕີຕີ້ສັນຍາລັກຂອງແອັດທີຣບີວຈະເປັນຮູບວົງມົນແອັດທຣິບີວໃດທີ່ຖືກໃຊ້ເປັນຄີຫຼັກຈະຖືກຂີດເສັ້ນກ້ອງກໍາກັບໄວ້.

emName

Position

emNo

EMPLOYEE

ຮູບທີ 5 ຮູບAttribute

* 1. ຄວາມສໍາພັນ (Relation)

ຄວາມສໍາພັນໃນທີ່ນີ້ໝາຍເຖິງຄວາມສໍາພັນລະຫວ່າງເອັນຕີຕີ້ໂດຍແຕ່ລະຄວາມສໍາພັນຄວນມີຊື່ລະບຸໄວ້ເພື່ອໃຊ້ອະທິບາຍເຊິ່ງປົກກະຕິຈະໃຊ້ສັນຍາລັກຮູບດອກຈັນທີ່ພາຍໃນລະບຸຄໍາກິລິຍາໄວ້ເພື່ອອະທິບາຍຄວາມສໍາພັນ.

SUBJECT

teaches

LUCTURE

M

1

ຮູບທີ 6 ຮູບAttribute

1. ຄວາມສໍາພັນແບບຢູນາຣີ(Unary Relationships):ເປັນຄວາມສໍາພັນລະຫວ່າງເອັນຕີຕີ້

ດຽວໂດຍຄວາມສໍາພັນແບບຢູນາຣີນີ້ກໍ່ຄືຄວາມສໍາພັນແບບຣີເຄີຊີບ (Recursive)ທີ່ເອີ້ນໃຊ້ໃນຕົວ

ນັ້ນເອງ.

manages

EMPLOYEE

1

M

ຮູບທີ 7 ຮູບAttribute

1. ຄວາມສໍາພັນແບບໄບນາຣີ (Binary Relationships):ການສໍາພັນຊະນິດນີ້ຈະມີເອັນຕີຕີ້ທີ່ກ່ຽວຂ້ອງກັນ2ເອັນຕີຕີ້ເຊິ່ງເປັນຄວາມສໍາພັນຊະນິດຫນຶ່ງທີ່ສາມາດພົບເຫັນໄດ້ຫຼາຍໃນຄວາມເປັນຈິງ.

MAJOR

live

FACULTY

M

1

ຮູບທີ 8 ຄວາມສໍາພັນແບບໄບນາຣີ *( Binary Relationships* )

1. ຄວາມສໍາພັນແບບເທີ້ນາຣີ (Ternary Relationships) ເປັນຄວາມສໍາພັນທີມກ່ຽວຂ້ອງ 3 ເອັນຕີຕີ້ດ້ວຍກັນ.

1

M

PATIENT

DOCTOR

write

Read

PRESCRIPTION

Appears in

output

ຮູບທີ 9 ຄວາມສໍາພັນແບບເທີ້ນາຣີ (Ternary Relationships)

* + - 1. ພາສາ NoSql (Not Only Structure Query Language)

1. NoSQL ແມ່ນຫຍັງ

  ຫຼາຍ​ຄົນ​ຄົງ​ພໍ​ຈະ​ເຄີຍ​ໄດ້ຍິນ​ກັນ​ມາແດ່ກ່ຽວກັບ​ເທັກ​ໂນ​ໂລ​ຢີ​ການ​ຈັດການ​ຂໍ້​ມູນ​ແບບ​ໃໝ່​ນີ້ ຊຶ່ງ​ກໍ​ຄື NoSQL ເມື່ອ​ເວົ້າ​ເຖິງ NoSQL ຈະ​ໄດ້ຍິນ​ຊື່​ເວັບ​ໄຊ​​ທີ່​ໃຫ່ຍ​ໆ ເຊັ່ນ Facebook,Twitter, FourSquare, Digg ແລະ ​ອື່ນ​ໆ.  ເຮັດໃຫ້​ເຮົາ​ຮັບ​ຮູ້​ວ່າ NoSQL ເປັນ​ລະບົບ​ຖານ​ຂໍ້​ມູນ​ສຳລັບ​ງານ​ທີ່​ຕ້ອງ​ຮອງ​ຮັບ​ຂໍ້​ມູນ​ຂະໜາດ​ໃຫ່ຍໆຮອງ​ຮັບ​ການ​ຂະຫຍາຍ​ລະບົບ​ໄດ້​ງ່າຍເປັນ​ຕົ້ນ

     ຊຶ່ງ​ກໍ​ເປັນ​ເຊັ່ນ​ນັ້ນ​ແທ້ ແຕ່​ງານ​ທີ່​ນ້ອຍ​ໆ ​ຈະ​ເຮັດ​ຢ່າງໃດ ໃຊ້​ງານ​ໄດ້​ບໍ່ ຄຸ້ມ​ຄ່າ​ທີ່​ຈະ​ນຳ NoSQL ມາ​ໃຊ້​ງານ​ຂະໜາດ​ນ້ອຍ​ ຫຼື ບໍ່ ຫຼື​ໃຊ້ Relational Database ກໍ​ພຽງ​ພໍ​ແລ້ວ ຄຳ​ຕອບ​ຄືຂຶ້ນ​ຢູ່​ກັບ​ລັກສະນະ​ໃນ​ການ​ນຳ​ມາ​ໃຊ້​ງານ ກ່ອນທີ່ຈະ​ຕອບ​ຄຳ​ຖາມວ່າ NoSQL ເປັນ​ຄຳ​ຕອບ​ຂອງ​ລະບົບ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຂອງ​ຫຼືບໍ່ ລອງ​ພິຈາລະນາ​ຫົວຂໍ້​ຕ່າງ​ໆດັ່ງ​ຕໍ່ໄປ​ນີ້

1. ຜູ້ໃຊ້ນັບມື້ນັບຫຼາຍ (BigUsers)

ຈະ​ເຫັນ​ໄດ້​ວ່າ​ໃນ​ຊ່ວງ​ເວລາ​ທີ່​ຜ່ານມາ​ນີ້ ແລະ ​ໃນ​ປັດ​ຈຸ​ບັນຜູ້​ທີ່​ໃຊ້​ງານ Internet ມີ​ແນວ​ໂນ້ມ​ຫລາຍ​ຂຶ້ນ​ເລື້ອຍ​ໆ ບໍ່​ວ່າ​ຈະ​ໃຊ້​ງານ​ຜ່ານ Desktop PC ຫລື Smartphone ຊຶ່ງ​ເທັກ​ໂນ​ໂລ​ຢີ​ຂອງ​ອຸປະກອນ (Devices) ມີ​​ຫຼາຍ​​ຂຶ້ນ ແລະ ​ໃຊ້​ງານ​ໄດ້​ງ່າຍ​ຂຶ້ນ

     ການ​ພັດທະນາ​ລະບົບ​ໃຫ້​ສາມາດ​ຮອງ​ຮັບ​ປະລິມານ​ການ​ເຂົ້າ​ໃຊ້​ງານ​ແຕ່​ລະ​ອຸປະກອນ (Devices) ເປັນ​ສິ່ງ​ໜຶ່ງ​ທີ່​ຕ້ອງ​ນຳ​ມາ​ພິຈາລະນາ  ແລະ ​ບໍ່​ພຽງ​ແຕ່​ຕ້ອງ​ຮອງ​ຮັບ​ການ​ເຂົ້າ​ໃຊ້​ງານ​ຂອງ​ຜູ້​ໃຊ້​ໄດ້​ເທົ່າ​ນັ້ນ  ເຮົາ​ຕ້ອງ​ຮອງ​ຮັບ​ວິທີ​ການ​ປ້ອນ​ຂໍ້​ມູນ​ແບບ​ໃໝ່ ຄື ແຕ່​ກ່ອນ​ຜູ້​ຈັດການ​ເນື້ອ​ຫາ​ຕ່າງ​ໆ ຄື Web Master, Web Editor, ຜູ້​ເບິ່ງ​ແຍງ​ລະບົບເປັນ​ຕົ້ນ ແຕ່​ປະຈຸບັນຜູ້​ທີ່​ປ້ອນ​ຂໍ້​ມູນ​ຄືຜູ້​ໃຊ້​ບໍລິການ (users) ໂດຍ​ກົງຜ່ານ​ອຸປະກອນ (Devices) ຕ່າງ​ໆ​ທີ່​ມີ​ຫຼາກ​ຫຼາຍ ແລະ ​ການ​ປ້ອນ​ຂໍ້​ມູນ​ກໍ​ງ່າຍ​ກວ່າ​ແຕ່​ກ່ອນ​ອີກດ້ວຍ

     ຍັງມີ​ປັດ​ໄຈ​ອື່ນ​ໆເຊັ່ນ ເທດ​ສະການ​ສຳຄັນ​ໆທີ່​ຄົນ​ຈະ​ເຂົ້າ​ມາ​ໃຊ້​ງານ​ຫລາຍ​ເປັນ​ພິເສດ ຫລື ​ຜູ້​ໃຊ້​ງານ​ທີ່​ບໍ່​ແມ່ນ​ແຕ່​ປະເທດ​ເຮົາ​ເທົ່າ​ນັ້ນ ເພາະ​ໂລກ​ອິນ​ເຕີເນັດ​ເຖິງ​ກັນ ອາດຈະ​ຕ້ອງ​ເບິ່ງ​ວ່າ​ລະບົບ​ເຮົາ​ມີ​ຜູ້​ເຂົ້າ​ໃຊ້​ງານ​ຈາກ​ຕ່າງ​ປະເທດ ຫລື​ ທົ່ວ​ໂລກ ​ຫລື​ ບໍ່​ອີກ​ດ້ວຍ

     ດັ່ງ​ນັ້ນ​ເຮົາ​ຕ້ອງ​ກັບ​ມາ​​ທົບທວນ​ວິທີ​ການ​ຈັດການ​ຖານ​ຂໍ້​ມູນ​ແລ້ວ​ວ່າ  ບໍ່​ແມ່ນ​ແຕ່​ເຮັດໃຫ້​ຮອງ​ຮັບ​ກັບ​ການ​ເຂົ້າ​ມາ​ໃຊ້​ງານ​ຂອງ​ຜູ້​ໃຊ້​ບໍລິການ​ເທົ່າ​ນັ້ນ ແຕ່​ຕ້ອງ​ຮອງ​ຮັບ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ຫລາຍ​ຂຶ້ນ​ເລື້ອຍ​ໆ ໄດ້​ອີກ

1. ປະເພດຂໍ້ມູນຕ່າງໆ ແລະ ຂໍ້ມູນທີ່ຕ້ອງການຈັດເກັບຫຼາຍຂຶ້ນເລື້ອຍໆ (BigData)

  ຈາກ​ຕັວ​ແປ​ຂອງ​ຜູ້​ໃຊ້​ງານ​ມີ​ຫລາຍ​ຂຶ້ນ​ເລື້ອຍ​ໆ ອຸປະກອນ​ໃນ​ການ​ເຂົ້າ​ໃຊ້​ງານ​ກໍ​ຫຼາກ​ຫຼາຍ ປະ​ເພດ​ຂອງ​ຂໍ້​ມູນ​ທີ່​ໄດ້​ຈາກ​ແຕ່​ລະ​ອຸປະກອນ​ກໍ​ຫຼາກ​ຫຼາຍ​ປະ​ເພດ ເຊັ່ນ ຂໍ້ຄວາມ,ຮູບ​ພາບ,ສຽງ,ວິ​ດີ​ໂອ,ຕຳແໜ່ງ​ສະຖານ​ທີ່ (GeoLocation) ແລະ​ອື່ນ​ໆ ແລະ ​ການ​ປ້ອນ​ຂໍ້​ມູນ​ເຫຼົ່າ​ນີ້​ກໍ​ງ່າຍ​ແສນ​ງ່າຍ ເພາະ​ເທັກ​ໂນ​ໂລ​ຢີ​ຂອງ​ຮາ​ດ​ແວ​ຣ ແລະ ​ຊອບແວ​​ມີ​ການ​ພັດທະນາ​ຫລາຍ​ຂຶ້ນ​ເລື້ອຍ​ໆ ໃຊ້​ງານ​ງ່າຍ​ຂຶ້ນສະ​ດວກ​ຂຶ້ ວ່ອງໄວ​ຂຶ້ນ​ເລື້ອຍ​ໆ

    ດັ່ງ​ນັ້ນ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ລັ່ງໄຫຼ​ເຂົ້າ​ມາ​ຈາກ​ອຸປະກອນ​ຕ່າງ​ໆ​ເຫຼົ່າ​ນີ້  ເຮົາ​ອາດຈະ​ຕ້ອງ​ນຳ​ມາ​​ວິ​ເຄາະ​ພຶດຕິກຳ​ຂອງ​ຜູ້​ໃຊ້​ບໍລິການ ການ​ສົ່ງ​ເສີມ​ກາ​ນຕະຫຼາດ ເຮັດ​ຂໍ້​ມູນ​ການ​ຕັດ​ສິນ​ໃຈ​ຂອງ​ຜູ້​ບໍລິຫານ ຂໍ້​ມູນ​ລູກ​ຄ້າ​ສຳພັນ ແລະ ​ອື່ນ​ໆ​ອີກ​ຫລວງ​ຫລາຍ  ການ​ນຳ​ລະບົບ​ຖານ​ຂໍ້​ມູນ​ແບບ​ເດີມ (Relational Database) ອາດຈະ​ບໍ່​ເໝາະ​ກັບ​ລັກສະນະ​ງານ​ບາງຢ່າງ​ອີກ​ຕໍ່ໄປ

1. ເຕັກໂນໂລຢີຮາດແວໄດ້ປ່ຽນແປງ ລາຄາກໍ່ຖືກຫຼຸດລົງແຕ່ວ່າປະສິດທິພາບດີຂື້ນ (Cloud Computing).

ເຮົາ​ອາດຈະ​ເລີ່ມ​ໄດ້ຍິນ​ຄຳ​ວ່າ Cloud Technology  ຊຶ່ງ​ກໍ​ມີ​ຫຼາຍ​ປະ​ເພດແຕ່​ໃນ​ທີ່​ນີ້ ເຮົາ​ຈະ​ເວົ້າ​ເຖິງ​ໃນ​ແງ່​ຂອງ​ການ​ນຳ​ມາ​ໃຊ້​ງານ  ໂດຍ​ປະຈຸບັນ​ຖ້າ​ໃຜ​ເຄີຍ​ໄດ້​ລອງ​ໃຊ້​ງານ EC2 ຂອງ Amazon ມາແດ່​ແລ້ວຈະ​ຮູ້​ຈັກ​ເປັນ​ຢ່າງ​ດີ​ວ່າການ​ຈະ​ມີ​ເຄື່ອງ Server ແຮງ​ໆ​ຈັກເຄື່ອງ ເປັນ​ເລື່ອງ​ທີ່​ງ່າຍ​ຫລາຍ​ ຫລື ການ​ຈະ​ມີ Server 10 ເຄື່ອງ ຫລື 20 ເຄື່ອງ ນຳ​ມາ​ຕໍ່​ເປັນ Database Cluster ນັ້ນ​ງ່າຍ​ຫລາຍ​ ພຽງແຕ່​ຄລິກ​ສ້າງ Instance ບໍ່​ຈັກ​ເທື່ອ​ກໍ​ໄດ້ Server ມາ​ໃຊ້​ງານ​ແລ້ວ ແລະ​ ລາ​ຄາ​ກໍ​ຖືກ​ຫລາຍ​ໆ ຖ້າ​ເຮົາ​ບໍ່​ໃຊ້​ງານ​ແລ້ວ ກໍ​ຍົກ​ເລີກ​ການ​ໃຊ້​ງານ ​ແລະ​ ຄືນ​ກັບ​ໄປ​ໄດ້​ທັນ​ທີ ເມື່ອ​ທຽບ​ກັບ​ສະໄໝ​ກ່ອນ​ທີ່​ເຮົາ​ຕ້ອງ​ການ​ມີ Server ຈັກ 10 ເຄື່ອງ ຈະ​ຕ້ອງ​ລົງ​ທຶນ​ຊື້​ເຄື່ອງ​ມາ​ຫລາຍ ຖ້າ​ເຊົາ​ໃຊ້​ງານ​ແລ້ວ ຕ້ອງ​ແບກ​ຮັບ​ພາລະ​ເຄື່ອງ Server ເຫຼົ່າ​ນີ້​ໄວ້​ຊຶ່ງ​ເປັນ​ຕົ້ນ​ທຶນ​ທີ່​ແພງ​ຫລາຍ​

     ຈາກ​ທີ່​ກ່າວ​ມາ​ຂ້າງ​ເທິງ ເຮົາ​ບໍ່​ໄດ້​ເນັ້ນ​ຂໍ້​ດີ​ຂອງ EC2 ແຕ່​ຢ່າງ​ໃດແຕ່​ກຳ​ລັງ​ຈະ​ເນັ້ນ​ວ່າ ພາ​ບລວມ​ຂອງ​ການ​ໃຊ້​ງານ Server ເລີ່ມ​ປ່ຽນ​ໄປຄື ໃຊ້​ງານ​ໄດ້​ງ່າຍ​ຂຶ້ນ ລາ​ຄາ​ຖືກ​ລົງ ແຕ່​ປະ​ສິດ​ທິ​ພາບ​ດີ​ຂຶ້ນ ຊຶ່ງ​ເປັນ​ສິ່ງ​ທີ່​ສຳຄັນ​ໃນ​ການ​ນຳ​ມາ​ພິຈາລະນາ​ຂອງ​ເທກ​ໂນ​ໂລ​ຢີ​ດ້ານ​ຖານ​ຂໍ້​ມູນ​ຄື ຖ້າ​ຕ້ອງ​ການ​ຈັດ​ເກັບ​ຖານ​ຂໍ້​ມູນ​ໃຫຍ່​ໆ ຫລື​ ຮອງ​ຮັບ​ຜູ້​ໃຊ້​ງານ​ໄດ້​ເປັນ​ຈຳນວນ​ຫລາຍ​ໆ ການ​ຂະຫຍາຍ​ລະບົບ​ຖານ​ຂໍ້​ມູນ​ເປັນ​ເລື່ອງ​ທີ່​ງ່າຍ​ຂຶ້ນ ຊຶ່ງ​ເຮັດ​ໄດ້​ໂດຍ​ການ​ເອົາ Server ມາ​ຕໍ່​ໆ​ກັນ​ອອກ​ໄປ ຫລື​ ເອີ້ນວ່າ​ການ​ຂະຫຍາຍ​ອອກ​ແນວ​ນອນ (Scale Out) ບໍ່​ແມ່ນ​ການ​ຂະຫຍາຍ​ລະບົບ​ຄືແຕ່​ກ່ອນ ຄື​ຂະຫຍາຍ​ອອກ​ແນວ​ຕັ້ງ (Scale Up) ແລະ​ ຕ້ອງ​ໃຊ້​ເຄື່ອງ Server ທີ່​ມີ​ປະ​ສິດ​ທິ​ພາບ​ສູງ​ໆ ຊຶ່ງ​ຈະ​ມີ​ຕົ້ນ​ທຶນ​ທີ່​ແພງ​ກວ່າ​ການ​ຂະຫຍາຍ​ແບບ​ແນວ​ນອນຫລາຍ​

     ດັ່ງ​ນັ້ນ​ການ​ຂະຫຍາຍ​ລະບົບ​ທີ່​ຢູ່​ເທິງ​ພື້ນ​ຖານ​ຂອງ NoSQL ຄື​ຮອງ​ຮັບ​ການ​ຂະຫຍາຍ​ລະບົບ​ແບບ​ແນວ​ນອນ (Scale Out) ຊຶ່ງ​ຈະ​ກະ​ຈາຍ​ຂໍ້​ມູນ​ໄປ​ເກັບ​ທີ່​ເຄື່ອງ Server ຫຼາຍ​ໆ​ເຄື່ອງ ແລະ ​ໃຊ້​ເຄື່ອງ Server ທົ້ວ​ໆ​ໄປ​ທີ່​ເອີ້ນວ່າ (Commodity Server) ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ໃຊ້ Server ທີ່​ເອີ້ນວ່າ Enterprise Server ທີ່​ມີ​ລາ​ຄາ​ແພງ​ຕາມ Spec ທີ່​ສູງ​ຂຶ້ນ​ເລື້ອຍ​ໆ ແລະ​ ການ​ບໍລິຫານ​ຈັດການ​ກໍ​ຍາກ​ຂຶ້ນ​ອີກ​ດ້ວຍ

1. ບັນຫາ​ຂອງ Relational Database

    ຖ້າ​ໃຜ​ທີ່​ໃຊ້​ງານ Relation Database ທີ່​ຕ້ອງ​ການ​ຮອງ​ຮັບ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຂະໜາດ​ໃຫຍ່​ໆ ຄົງ​ຫຼີກລ້ຽງບໍ່ໄດ້ເລື່ອງ​ການເຮັດ Sharding ແລະ Distributed Cache ເພາະ​ເປັນ​ຕັວ​ຫຼັກ​ທີ່​ຕ້ອງ​ເຮັດ ເພື່ອ​ຂະຫຍາຍ​ລະບົບ​ຖານ​ຂໍ້​ມູນ​ຂອງ Relational Database ໃຫ້​ສາມາດ​ຮອງ​ຮັບ​ຂໍ້​ມູນ​ທີ່​ຫລາຍ​ຂຶ້ນ ແລະ​ຮອງ​ຮັບ​ຈຳນວນ​ການ​ເຂົ້າ​ມາ​ໃຊ້​ງານ​ລະບົບ​ໄດ້​ຫລາຍ​ຂຶ້ນ

1. Manual Sharding

    ການ​ແບ່ງ​ຕາ​ຕະລາງ​ຖານ​ຂໍ້​ມູນ (Table) ອອກ​ເປັນ​ສ່ວນ​ໆ ແລ້ວ​ກໍ່ການ​ກະ​ຈາຍ​ໄປ​ຈັດ​ເກັບ​ໃນ​ຫຼາຍ​ໆ Server  ເພື່ອ​ໃຫ້​ແຕ່​ລະ​ຕາ​ຕະລາງ (Table) ຂອງ​ຖານ​ຂໍ້​ມູນ​ບໍ່​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ຫຼາຍ​ເກີນ​ໄປ ເພາະ​ຖ້າ​ຂໍ້​ມູນ​ໃນ​ແຕ່​ລະ​ຖານ​ຂໍ້​ມູນ​ຫູາຍ​ເກີນ​ໄປ ຈະ​ເຮັດໃຫ້​ລະບົບ​ຖານ​ຂໍ້​ມູນ​ຊ້າ​ ແຕ່​ບັນຫາ​ກໍ​ຈະ​ຕາມ​ມາ​ອີກ​ຄື ເມື່ອ​ຕ້ອງ​ກະ​ຈາຍ​ຂໍ້​ມູນ​ອອກ​ໄປ​ໃນ​ແຕ່​ລະ Server ການ​ຈະ​ ​ກັບ​ຂໍ້​ມູນ ເຊັ່ນ ເພີ່ມ, ແກ້​ໄຂ, ລົບ, ດຶງ​ຂໍ້​ມູນ​ມາ​ສະແດງ​ຕ່າງ​ໆ ຈະ​ຕ້ອງ​ເຮັດ​ຜ່ານ Application ຫຼື ​ຕ້ອງ​ມີ Server ບາງ​ໂຕ​ທີ່​​ຖ້າ​ດຶງ​ຂໍ້​ມູນ​ແຕ່​ລະ Server ມາ​ທັງໝົດ​ເປັນ​ກ້ອນ​ດຽວ ນັ້ນ​ໝາຍ​ຄວາມ​ວ່າ ເຮົາ​ຕ້ອງ​ເຮັດ​ດ້ວຍ​ໂຕ​ເຮົາ​ເອງ ບໍ່​ແມ່ນ​ລະບົບ​ຖານ​ຂໍ້​ມູນ​ຈັດການ​ໃຫ້ (Manual Sharding)

1. Distributed Cache

    ເມື່ອ​ເຮົາ​ຕ້ອງ​ການ​ໃຫ້​ລະບົບ​ຮອງ​ຮັບ​ການ​ເຂົ້າ​ມາ​ໃຊ້​ງານ​ຫລາຍ​ໆ​ໄດ້​ນັ້ນ ຖ້າ​ຈະ​ຕ້ອງ​ເຂົ້າ​ມາ​ອ່ານ​ຂໍ້​ມູນ​ຜ່ານ Database ໂດຍ​ກົງ​ມັນອາດຈະ​ຮອງ​ຮັບ​ບໍ່​ໄຫວ ຫຼື ​ເຮັດ​ໄດ້​ຊ້າ ດັ່ງ​ນັ້ນ​ຈະ​ຕ້ອງ​ມີ​ການເຮັດ Cache Layer ຂຶ້ນ​ມາ ຄື​ແທນ​ທີ່​ຈະ​ເຂົ້າໄປ​ອ່ານ​ຈາກ​ຖານ​ຂໍ້​ມູນ​ໂດຍ​ກົງ ກໍ​ໃຫ້​ອ່ານ​ຜ່ານ Cache ກ່ອນ ດັ່ງ​ນັ້ນ​ການ​ອ່ານ​ຂໍ້​ມູນ​ຈາກ Cache ເປັນ​ການ​ອ່ານ​ຈາກ Memory ໂດຍ​ກົງ ເຮັດໃຫ້​ຮອງ​ຮັບ​ປະລິມານ​ການ​ເຂົ້າ​ມາ​ໃຊ້​ງານ​ໄດ້​ຫຼາຍ​ຂຶ້ນ

    ແຕ່​ບັນຫາ​ຄື ການເຮັດ Cache Layer ນີ້ ຮອງ​ຮັບ​ສະເພາະ​ການ​ອ່ານ​ຂໍ້​ມູນ​ເທົ່າ​ນັ້ນ ບໍ່​ຮອງ​ຮັບ​ການ​ຂຽນ​ຂໍ້​ມູນ​ໄດ້ ຖ້າ​ຕ້ອງ​ການ​ຮອງ​ຮັບ​ການ​ຂຽນ​ຂໍ້​ມູນ​ປະລິມານ​ຫຼາຍ​ໆ ແລະ​ອ່ານ​ຂໍ້​ມູນ​ປະລິມານ​ຫຼາຍ​ໆ ຈຶ່ງ​ເປັນ​ສິ່ງ​ທີ່ Relational Database ບໍ່​ສາມາດ​ຮອງ​ຮັບ​ງານ​ໃນ​ລັກສະນະ ອ່ານ​,​ຂຽນ ຂໍ້​ມູນ​ປະລິມານ​ຫຼາຍ​ໆໄດ້​ດີ​ ແລະ​ ສິ່ງ​ສຳຄັນການເຮັດ Cache Layer ຈະ​ຕ້ອງ​ມີ​ການ​ດູແລຮັກສາ ແລະ​ ໃຊ້ Server ແຍກ​ອອກ​ໄປ​ຕ່າງ​ຫາກ​ອີກ​

    ຈາກ​ຈຸດ​ນີ້​ເອງ ທັງ​ການ​ເຮັດ Sharding ແລະ Caching ເປັນ​ສິ່ງ​ທີ່​ຖືກ​ພັດທະນາ​ຂຶ້ນ​ໃນ NoSQL ເທ​​ກ​ໂນ​ໂລ​​ຢີ  ໂດຍ​ຮອງ​ຮັບ Auto-Sharding ແລະ Integrated Caching ໃນ​​ຕັວ​ເອງ ດັ່ງນັ້ນ​ເຮົາ​ຈຶ່ງ​ໄດ້​ເຫັນ NoSQL ຖືກ​ນຳໄປ​ໃຊ້​ງານ​ກັບ​ລະບົບ​ໃຫຍ່ໆ ເຊັ່ນ Facebook,Twitter, FourSquare, Digg ແລະ​ອື່ນໆ ເພ​​າະ NoSQL ອອກ​ແບບ​ມາ​ເພື່ອ​ຮອງ​ຮັບ​ຄວາມ​ຕ້ອງການ​ງານ​ໃຫຍ່ໆ​ໄດ້​ດີ​ໂດຍ​ສະເພາະ​ຢູ່​ແລ້ວ ແຕ່​ເຖິງ​ຢ່າງ​ໃດກໍ​ຕາມ​ຍັງ​ມີ​ຄຸນສົມບັດ​ອື່ນໆ ທີ່​ໜ້າ​ສົນໃຈ​ໃນ NoSQL ເທ​​ກ​ໂນ​ໂລ​​ຢີ

1. ຄຸນສົມບັດ​ຂອງ NoSQL Database

* Dynamic Schemas

    ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຕ່າງ​ໆ ໃນ​ຖານ​ຂໍ້​ມູນ​ແບບ Relational Database ເຮົາ​ຈະ​ຕ້ອງ​ມີ​ການສ້າງ Schema ຫຼື ຮູບ​ແບບ​ຂອງ​ໂຄງ​ສ້າງ​ຕາ​ຕະລາງວ່າ​ຈະ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຫຍັງ​ ເມື່ອ​ຕ້ອງ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ເພິ່ມເຕີມ​ຕ້ອງ​ປ່ຽນ Schema ພາຍ​ຫຼັງ (Alter-Table) ກ່ອນ​ຈະ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຮູບ​ແບບ​ໃໝ່​ໄດ້

    ແຕ່​ໃນ​ປະຈຸບັນ ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ມີ​ການ​ປ່ຽນ​ແປງ​ຕະຫລອດ​ເວລາ ເພາະ​ຄວາມ​ຕ້ອງ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຕ່າງ​ໆ ມີ​ຫຼາກ​ຫຼາຍ​​ຂຶ້ນ​ເລື້ອຍ​ໆ ການ​ກຳນົດ​ໂຄງ​ສ້າງ​ຂອງ​ຕາ​ຕະລາງ​ຖານ​ຂໍ້​ມູນ ຫຼື ການ​ຕ້ອງ​ປ່ຽນ​ໂຄງ​ສ້າງ​ຖານ​ຂໍ້​ມູນ​ເລື້ອຍໆ ໂດຍ​ທີ່​ຂໍ້​ມູນ​ຍັງມີ​ຢູ່​ແລ້ວ ເປັນ​ເລື່ອງ​ທີ່​ຍາກ​ຫຼາຍ ​ຫຼື ເຮັດ​ບໍ່​ໄດ້​ເລີຍ ວິທີ​ການ​ຄື​ອາດ​ຕ້ອງ​ແຍກ​ອອກ​ເປັນ​ຕາ​ຕະລາງ​ໃໝ່ ຊຶ່ງ​ເປັນ​ວິທີ​ແກ້​ບັນຫາ​ຊົ່ວ​ຄາວ​ເທົ່າ​ນັ້ນ

    ລະບົບ​ຖານ​ຂໍ້​ມູນ​ແບບ NoSQL  ເຮົາ​ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ມີ Schema ທີ່​ຕາຍ​ຕົວ ຫຼື ​ບໍ່​ຕ້ອງ​ມີ Schema ກ່ອນທີ່ຈະ​ຈັດ​ເກັບ​ຂໍ້​ມູນ ຂໍ້​ມູນ​ແຕ່​ລະ​ແຖວ ສາມາດ​ຈັດ​ເກັບ​ໄດ້​ຕາມ​ຕ້ອງ​ການ ຈະ​ເພີ່ມ​ ຫຼື ​ຫຼຸດ ກໍ​ບໍ່​ມີ​ບັນຫາ​ກັບ​ລະບົບ  ເຮັດໃຫ້​ເຮົາ​ສາມາດ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ໄດ້​ຕາມ​ທີ່​ຕ້ອງ​ການ ປ່ຽນ​ແປງ​ໄດ້​ຕະຫລອດ​ເວລາ ສະ​ດວກ​ ແລະ ວ່ອງໄວ

* Auto-Sharding

    ເມື່ອ​ຂໍ້​ມູນ​ມີ​ຂະໜາດ​ໃຫຍ່ ຫຼື ​ເຮົາ​ຕ້ອງ​ການ​ເພີ່ມ​ປະ​ສິດ​ທິ​ພາບ​ການ ອ່ານ​ ແລະ ຂຽນ​ຂໍ້​ມູນ​ປະລິມານ​ຫຼາຍ​ໆ ການເຮັດ Sharding ໃນ​ລະບົບ NoSQL Database ຈະ​ກໍ່ການ​ກະ​ຈາຍ​ຂໍ້​ມູນ​ໄປ​ຍັງ Server ຕ່າງ​ໆ​ອັດ​ໂນ​ມັດ​​ (Auto-Sharding) ຜູ້​ພັດທະນາ (Developer) ບໍ່​ຕ້ອງ​ຂຽນ​ໂປຣ​ແກຣມ​ໃນ​ການ​ກະ​ຈາຍ​ຂໍ້​ມູນ​ເອງຄືກັບ Relational Database  
    ການ​ກະ​ຈາຍ​ຂໍ້​ມູນ​ອອກ​ໄປ​ຫຼາຍ​ໆ Server ນີ້​ຍັງ​ເຮັດໃຫ້​ມີ​ຂໍ້​ດີ​ຄື ປະ​ຢັດ​ຕົ້ນ​ທຶນ​ໃນ​ການ​ຂະຫຍາຍ​ລະບົບ ເພາະ​ເປັນ​ການ​ຂະຫຍາຍ​ແບບ​ແນວ​ນອນ (Scale Out) ຊຶ່ງ​ສາມາດ​ນຳ Server ປົກກະຕິ​ທົ່ວ​ໄປ​ມາ​ໃຊ້​ງານ​ໄດ້ ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ເປັນ Enterprise Server

* Replication

    ການ​ສຳ​ເນົາ​ຂໍ້​ມູນ​ຈາກ​ເຄື່ອງ​ໜຶ່ງ​ໄປ​ຢັງ​ອີກ​ເຄື່ອງ​ໜຶ່ງ (Replication) ເມື່ອ Server ໜຶ່ງ​ເສຍ​ຫາຍ ອີກ​ເຄື່ອງ​ໜຶ່ງ​ຈະ​ຂຶ້ນ​ມາ​ເຮັດວຽກ​ແທນ​ທັນ​ທີ ໂດຍ​ຂໍ້​ມູນ​ຂອງ​ແຕ່​ລະ​ເຄື່ອງ​ຈະ​ມີ​ຂໍ້​ມູນ​ຄືກັນ ດັ່ງ​ນັ້ນ Replication ເປັນ​ໜຶ່ງ​ຄຸນສົມບັດ​ທີ່​ຕອບສະໜອງ​ຕໍ່​ການ​ໃຊ້​ງານ​ທີ່​ຕ້ອງ​ການ​ຄວາມ​ຕໍ່​ເນື່ອງ​ໄດ້​ຕະຫລອດ​ເວລາ (High Availability)

* Integrated Caching

    ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ໃຊ້​ງານ​ເລື້ອຍໆ ເຂົ້າ​ໄວ້​ໃນ Memory (RAM) ຊຶ່ງ​ເປັນ​ຄຸນສົມບັດ​ເດັ່ນ​ຂອງ NoSQL ທີ່​ທັງຫມົດ Caching ໄວ້​ໃນ​ຕົວ​ເອງ​ຢູ່​ແລ້ວ ເຮົາ​ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ເຮັດ Cache Layer ຄືກັບ Relational Database ອີກ​ຕໍ່ໄປ ທີ່​ຕ້ອງ​ເຮັດ Cache Layer ແຍກ​ຕ່າງ​ຫາກ ​ແລະ​ ເບິ່ງແຍງຮັກສາ​ແຍກ​ອອກ​ໄປ​ຕ່າງ​ຫາກ​ອີກ​ດ້ວຍ

* ປະ​ເພດ​ຂອງ​ຖານ​ຂໍ້​ມູນ NoSQL

    NoSQL ຖືກ​ແບ່ງ​ປະ​ເພດ​ຕາມ​ລັກສະນະ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ແຕກ​ຕ່າງ​ກັນ ດັ່ງ​ນັ້ນ​ການ​ຈະ​ເລືອກ NoSQL Database ໂຕໃດໂຕໜຶ່ງ ຈະ​ຕ້ອງ​ເບິ່ງ​ອີກ​ວ່າ ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຂອງ​ຖານ​ຂໍ້​ມູນ​ເປັນ​ແບບໃດ ເຊັ່ນ

- Document databases ເຊັ່ນ MongoDB, CouchDB, Elasticsearch

- Graph stores ເຊັ່ນ Neo4J, Infinite Graph, InfoGrid

- Key-value stores ເຊັ່ນ DynamoDB, Redis, MemcacheDB

- Wide-column stores ເຊັ່ນ Cassandra, Amazon SimpleDB, Hadoop / HBase

* Open source License

    ໂດຍ​ສ່ວນ​ໃຫຍ່​ແລ້ວ NoSQL ຈະ​ເປັນ​ລິຂະ​ສິດ​ແບບ Open source ຊຶ່ງ​ບໍ່​ຕ້ອງ​ເສຍ​ຄ່າ​ໃຊ້​ຈ່າຍ​ໃນ​ການ​ນຳ​ມາ​ໃຊ້​ງານ ດັ່ງ​ນັ້ນ​ເຮົາ​ສາມາດ​ນຳ NoSQL Database ແຕ່​ລະ​ຕົວ​ມາ​ຕິດ​ຕັ້ງ​ໃຊ້​ງານ​ໄດ້​ໂດຍ​ບໍ່​ເສຍ​ຄ່າ​ໃຊ້​ຈ່າຍ​ໃດ​ໆ (ຟ​ຣີ)

1. ນຳ NoSQL ມາ​ໃຊ້​ງານ​ຂະໜາດ​ນ້ອຍ​ໄດ້​ ຫຼື​ ບໍ່

    ຈາກ​ທີ່​ກ່າວ​ມາ​ແລ້ວ ຄົງ​ພໍ​ຈະ​ຕອບ​ຄຳ​ຖາມ​ນີ້​ໄດ້​ວ່າ ການ​ນຳ NoSQL Database ​ເມື່ອ​ນຳ​ມາ​ໃຊ້​ໃນ​ງານ​ຂະໜາດ​ໃຫຍ່​ນັ້ນ ເໝາະ​ສົມ​ຢ່າງ​ແນ່ນອນ ແຕ່​ຖ້າ​ເປັນ​ລະບົບ​ທົ່ວ​ໆ​ໄປ​ ຄວນ​ຈະ​ນຳ NoSQL ມາ​ໃຊ້​ງານ ​ຫຼື ​ບໍ່

    ຄຳ​ຕອບ​ຄື ຂຶ້ນ​ຢູ່​ກັບ​ລັກສະນະ​ງານ​ວ່າ ເຮົາ​ຈະ​ໃຊ້​ຄຸນສົມບັດ​ຫຍັງ​ຂອງ NoSQL ຖ້າ​ເຮົາ​ຕ້ອງ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ບໍ່​ຕ້ອງ​ຢຶດ​ຕິດ​ກັບ​ໂຄງ​ສ້າງ (Dynamic Schema) ແລະ ​ຕ້ອງ​ການ​ເຂົ້າ​ໃຊ້​ງານ​ລະບົບ​ທີ່​ວ່ອງໄວ (Integrated Caching) ຂໍ້​ມູນ​ອາດຈະ​ຢັງ​ບໍ່​ຫຼາຍ​ເທົ່າ​ໃດ ເຈົ້າ​ອາດຈະ​ໃຊ້ NoSQL ໄດ້​ຢ່າງ​ແນ່ນອນ

    ແຕ່​ຖ້າ​ບໍ່​ຕ້ອງ​ການ​ໃຊ້​ງານ (Dynamic Schema) ບໍ່​ຕ້ອງ​ການ​ເຂົ້າ​ໃຊ້​ງານ​ທີ່​ວ່ອງໄວ (Integrated Caching) ເພາະ​ໃຊ້ Relational Database ກໍ​ເຮັດ​ໄດ້​ດີ​ຢູ່​ແລ້ວ Database Server ກັບ Web Server ກໍ​ຢູ່​ທີ່ Server ດຽວ​ກັນ ຂໍ້​ມູນ​ບໍ່​ຫລາຍ​ນັ້ນ ບໍ່​ຕ້ອງ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ເພີ່ມ​ຂະຫຍາຍ​ຂຶ້ນ​ທຸກ​ມື້​ໆ ຜູ້​ເຂົ້າ​ໃຊ້​ງານ​ກໍ​ບໍ່​ໄດ້​ຫລາຍ ເບິ່ງ​ແລ້ວ​ວ່າ​ລະບົບ​​ບໍ່​ມີ​ແນວ​ໂນ້ມ​ຈະ​ຕ້ອງ​ຂະຫຍາຍ​ລະບົບ​ໃນ​ອະນາຄົດ​ອັນ​ໃກ້ ເຈົ້າ​ສາມາດ​ໃຊ້​ງານ Relational Database ໄດ້​ດີ​ຢູ່​ແລ້ວ ຢ່າງ​ບໍ່​ມີ​ບັນຫາ

* + - 1. ລະບົບຈັດການຖານຂໍ້ມູນ (Database Management Systems: DBMS)
    1. ພາສາທີ່ໃຊ້ຂຽນໂປຣແກຣມ
       1. ໂປຣແກຣມ Visual Studio Code
       2. ພາສາ JavaScript
       3. Reactjs
       4. Nodejs
       5. ExpressJS
       6. Git ແລະ GitHub
       7. GraphQL API

ວັນທີ ……/……./……… ວັນທີ ……/……./……… ວັນທີ ……/……./………

ລາຍເຊັນຄະນະກໍາມະການ ລາຍເຊັນອາຈານທີ່ປຶກສາ ລາຍເຊັນນັກສຶກສາ