

ជំពូកទី ៦

Normalization

នៅពេលយើងចាប់ផ្តើមធ្វើការកសាង database សំរាប់ relational system គោលដៅជាចំបងក្នុងការអភិវឌ្ឍន៍ logical data model គឺបង្កើតការបង្ហាញដ៏ត្រឹមត្រូវចំពោះទិន្នន័យ, relationship ព្រមទាំង constraint ផងដែរ ដូច្នេះដើម្បីសំរេចបានគោលដៅនេះ យើងត្រូវតែកំណត់អោយបាននូវសំនុំនៃ relations ដ៏សមរម្យ ហើយបច្ចេកទេសដែលយើងអាចប្រើសំរាប់ជួយក្នុងការកំណត់សំនុំ relations ដ៏ត្រឹមត្រូវគេហៅថា Normalization ។

1. The Purpose of Normalization

Normalization A technique for producing a set of relations with desirable properties, given the data requirements of an enterprise.

Normalization គឺជាបច្ចេកទេសមួយក្នុងការបង្កើតសំនុំ relations ដោយមាន properties ដែលគេត្រូវការ ទៅតាមតំរូវការទិន្នន័យរបស់ enterprise ។

កិច្ចដំណើរការ normalization ដំបូងត្រូវបានអភិវឌ្ឍន៍ឡើងដោយ E.F.Codd ក្នុងឆ្នាំ 1972 ។ Normalization ជាទូទៅធ្វើការត្រួតពិនិត្យជាសេរីទៅលើ relations ដើម្បីកំណត់ថា តើ relations បានបំពេញ រឺល្មើសលើតំរូវការនៃ Normal Form ។ Normal Form 3 ដំបូង ដែលគេតែងតែប្រើប្រាស់ញឹកញាប់ជាងគេគឺ First Normal Form (1NF), Second Normal Form (2NF) និង Third Normal Form (3NF) បន្ទាប់មកគឺ BCNF (Boyce-Codd Normal Form) បង្កើតដោយ R.Boyce and E.F.Codd ក្នុងឆ្នាំ 1974 ហើយ Normal Form ទាំងនេះមានមូលដ្ឋានលើអាស្រ័យអនុគមន៍ (Functional dependencies) ក្នុងចំណោម attributes នៃ relation ។ Higher Normal Form បន្ទាប់ពី BCNF ដែលមានការកំរិតខ្លាំងជាងគឺ Fourth Normal Form (4NF) បង្កើតដោយ Fagin ក្នុងឆ្នាំ 1977 និង Fifth Normal Form (5NF) បង្កើតដោយ Fagin ក្នុងឆ្នាំ 1979 ពុំសូវបានគេប្រើប្រាស់ប៉ុន្មានទេ ។

យើងដឹងហើយថា relation មួយមានផ្ទុកនូវបណ្តុំ attributes ជាច្រើន ហើយ relational schema ផ្ទុកនូវបណ្តុំ relations ជាច្រើន ដូច្នេះ attributes ប្រហែល

ជាត្រូវគេផ្តល់បញ្ហាដើម្បីបង្កើតជា relational schema ដោយយោលទៅលើគោលការ ក្នុងការកសាង database ដោយបំប្លែងពី ER diagram ទៅកាន់ relations ។

2. Update Anomalies

គោលបំណងចំបងក្នុងការកសាង relational database គឺធ្វើការប្រមូលផ្តុំ attributes បញ្ចូលគ្នាបង្កើតជាសំណុំ relations ជាច្រើន ដើម្បីបន្ថយភាពស្ងួនទិន្នន័យអោយកាន់តែតិច (minimize data redundancy) ដោយហេតុនេះវាក៏បានកាត់បន្ថយទំហំរក្សាទុកទិន្នន័យផងដែរ (reduce storage file space) នៅពេលកសាងជាមួយ DBMS ។ បញ្ហាដែលពាក់ព័ន្ធជាមួយ ភាពស្ងួនទិន្នន័យ (data redundancy) បង្ហាញការប្រៀបធៀបតាមរយៈ Staff_Branch relation រូប 6.1 និង Staff and Branch relations រូប 6.2 ។

ឧបមាថា យើងមាន Staff_Branch relation ដូចខាងក្រោម

Staff_Branch (Staff_No, SName, SAddress, Position, Salary, Branch_No, BAddress, Tel_No)

Staff_No	SName	SAddress	Position	Salary
S21	Ny Tony	19 Sothearos St, PP	Manager	30000
S37	Ann Ford	34 Monivong St, PP	Snr Asst	12000
S14	Khun Vitou	234 Tonele Mekong St, PP	Deputy	18000
S9	Phal Soheat	13 Tonele Mekong St, PP	Assistant	9000
S5	Pak Vannly	17 Sihanouk St, PP	Manager	24000
S41	Thay Sokan	37 Sihanouk St, PP	Assistant	9000

Branch_No	BAddress	Tel_No
B5	22 Chak Anre St, PP	023 217272
B3	163 Chba Ampov St, PP	023 210003
B3	163 Chba Ampov St, PP	023 210003
B7	72 Chak Anre St, PP	023 725456
B3	163 Chba Ampov St, PP	023 210003
B5	22 Chak Anre St, PP	023 217272

Staff_Branch relation (រូប 6.1)

តាមរយៈរូបខាងលើ យើងសង្កេតឃើញថា Staff_Branch relation មានផ្ទុកទិន្នន័យស្ងួន ព្រោះព័ត៌មានលំអិតនៃ branch (details of branch) បង្ហាញម្តងហើយម្តងទៀតចំពោះរាល់ សមាជិក staff ដែលស្ថិតនៅក្នុង branch ណាមួយជាក់លាក់ ប៉ុន្តែតាមពិតទៅព័ត៌មានលំអិតនៃ branch ត្រូវបង្ហាញឡើងតែម្តងគត់ចំពោះ branch នីមួយៗ ហើយមានតែ branch number (Branch_No) ទេដែលត្រូវបង្ហាញម្តងហើយម្តងទៀតក្នុង Staff relation ។

Relations ដែលមានភាពស្មុន់ស្មាត អាចមានបញ្ហាគេហៅថា Update Anomalies ដែលចែកជា 3 គឺ Insertion, Deletion or Modification Anomalies ។

Update Anomalies គឺជាបញ្ហាដែលធ្វើអោយ database មានភាពមិនប្រក្រតីនៅពេលដែល database យើងមានផ្ទុកភាពស្មុន់ស្មាត។

2.1. Insertion Anomalies

Insertion anomalies គឺជាបញ្ហាដែលកើតមាននៅពេលដែលយើងបញ្ចូលទិន្នន័យ។ តាមរយៈរូប 6.1 មានបញ្ហា 2 ជាចំបងចំពោះ Insertion anomalies ដូចខាងក្រោម:

- នៅពេលបញ្ចូលព័ត៌មានលំអិតចំពោះបុគ្គលិកថ្មីទៅក្នុង Staff_Branch relation យើងក៏ត្រូវតែបញ្ចូលព័ត៌មានលំអិតនៃ Branch ដែល staff នោះស្ថិតនៅផងដែរ។ ឧទាហរណ៍: បញ្ចូលបុគ្គលិកថ្មីម្នាក់ទៅកាន់ branch number B7 នោះយើងត្រូវតែបញ្ចូលព័ត៌មានលំអិតនៃ branch នោះអោយបានត្រឹមត្រូវ បើពុំដូច្នោះទេ database យើងនឹងមានភាពមិនស៊ីសង្វាក់គ្នានៃទិន្នន័យ (inconsistency) កើតឡើង។

- នៅពេលបញ្ចូលព័ត៌មានលំអិតចំពោះ Branch ដែលបច្ចុប្បន្នមិនទាន់មានសមាជិកក្នុង Staff_Branch relation យើងចាំបាច់ត្រូវតែបញ្ចូលតំលៃ Null ទៅកាន់ attributes ទាំងអស់នៃ Staff ដូចជា Staff_No ជាដើម ប៉ុន្តែ Staff_No មានតួនាទីជា primary key ដូច្នេះបំណងបញ្ចូលតំលៃ Null វាបានល្មើសទៅលើ entity constraints ពោលជាទូទៅ Null មិនត្រូវបានអនុញ្ញាតិអោយបញ្ចូលទេជាដាច់ខាត។

2.2. Deletion Anomalies

Deletion anomalies គឺជាបញ្ហាដែលកើតមាននៅពេលដែលយើងលុបទិន្នន័យ។

ប្រសិនបើយើងលុប staff ណាមួយចេញពី Staff_Branch relation ដែលជាសមាជិកចុងក្រោយគេបង្អស់ (the last member) រឺ branch នោះមានសមាជិកតែម្នាក់គត់ នោះព័ត៌មានលំអិតដែលពាក់ព័ន្ធនឹង branch នោះក៏ត្រូវបានលុបចាត់ចេញពី database ផងដែរ។ ឧទាហរណ៍: ប្រសិនបើយើងលុប staff number S9 (Phal Soheat) ចេញពី Staff_Branch relation នោះព័ត៌មានលំអិតដែលពាក់ព័ន្ធនឹង Branch_No B7 ក៏ត្រូវបានលុបចាត់ចេញពី database ផងដែរ។

2.3. Modification Anomalies

Modification anomalies គឺជាបញ្ហាដែលកើតមាននៅពេលដែលយើងកែប្រែទិន្នន័យ។

ប្រសិនបើយើងធ្វើការកែប្រែតំលៃ attribute ណាមួយនៃ Branch ណាមួយក្នុង Staff_Branch relation ឧទាហរណ៍ Telephone number នៃ Branch number B3 នោះយើងក៏ត្រូវតែកែប្រែរាល់ rows ដែល staffs ទាំងនោះស្ថិតនៅ branch នោះផងដែរ។ ប្រសិនបើការកែប្រែពុំបានអនុវត្តលើ rows ទាំងអស់នៃ Staff_Branch relation ពេលនោះ database នឹងមាន inconsistency កើតឡើង។

ដូច្នេះដើម្បីលុបបំបាត់ Update Anomalies គេត្រូវធ្វើការបំបែក Staff_Branch relation ទៅជា 2 relations ដូចខាងក្រោម

Staff (Staff_No, SName, SAddress, Position, Salary, Branch_No)

Branch (Branch_No, BAddress, Tel_No)

Staff_No	SName	SAddress	Position	Salary	Branch_No
S21	Ny Tony	19 Sothearos St, PP	Manager	30000	B5
S37	Ann Ford	34 Monivong St, PP	Snr Asst	12000	B3
S14	Khun Vitou	234 Tonele Mekong St, PP	Deputy	18000	B3
S9	Phal Soheat	13 Tonele Mekong St, PP	Assistant	9000	B7
S5	Pak Vannly	17 Sihanouk St, PP	Manager	24000	B3
S41	Thay Sokan	37 Sihanouk St, PP	Assistant	9000	B5

Staff relation

Branch_No	BAddress	Tel_No
B3	163 Chba Ampov St, PP	023 210003
B5	22 Chak Anre St, PP	023 217272
B7	72 Chak Anre St, PP	023 725456

Branch relation (រូប 6.2)

3. Functional Dependency

ដំណើរការ normalization មានមូលដ្ឋានគ្រឹះទៅលើ functional dependencies។ Functional Dependency (អាស្រ័យអនុគមន៍) គឺជាការចងទំនាក់ទំនងរវាង attributes 2 (Functional Dependency is a particular relationship between two attributes)។ ឧទាហរណ៍ A និង B ជា attributes នៃ relation R ប្រសិនបើ B is functionally dependent on A (តាងដោយសញ្ញា $A \longrightarrow B$) លុះត្រាតែ បើយើងដឹងតំលៃ A នោះយើងក៏អាចដឹងតំលៃរបស់ B បានដែរ។

ឧទាហរណ៍: Students (StudentID, StudentName, Sex, Address)

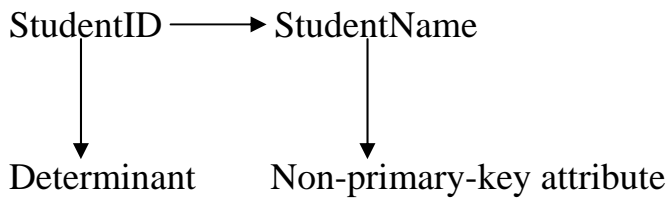
StudentID \longrightarrow StudentName

StudentID \longrightarrow Sex

StudentID \longrightarrow Address

Attribute ដែលស្ថិតផ្នែកខាងឆ្វេងនៃនិមិត្តសញ្ញាប្រញូញ (\longrightarrow) នៅក្នុង Functional dependency គេហៅថា **Determinant** ។

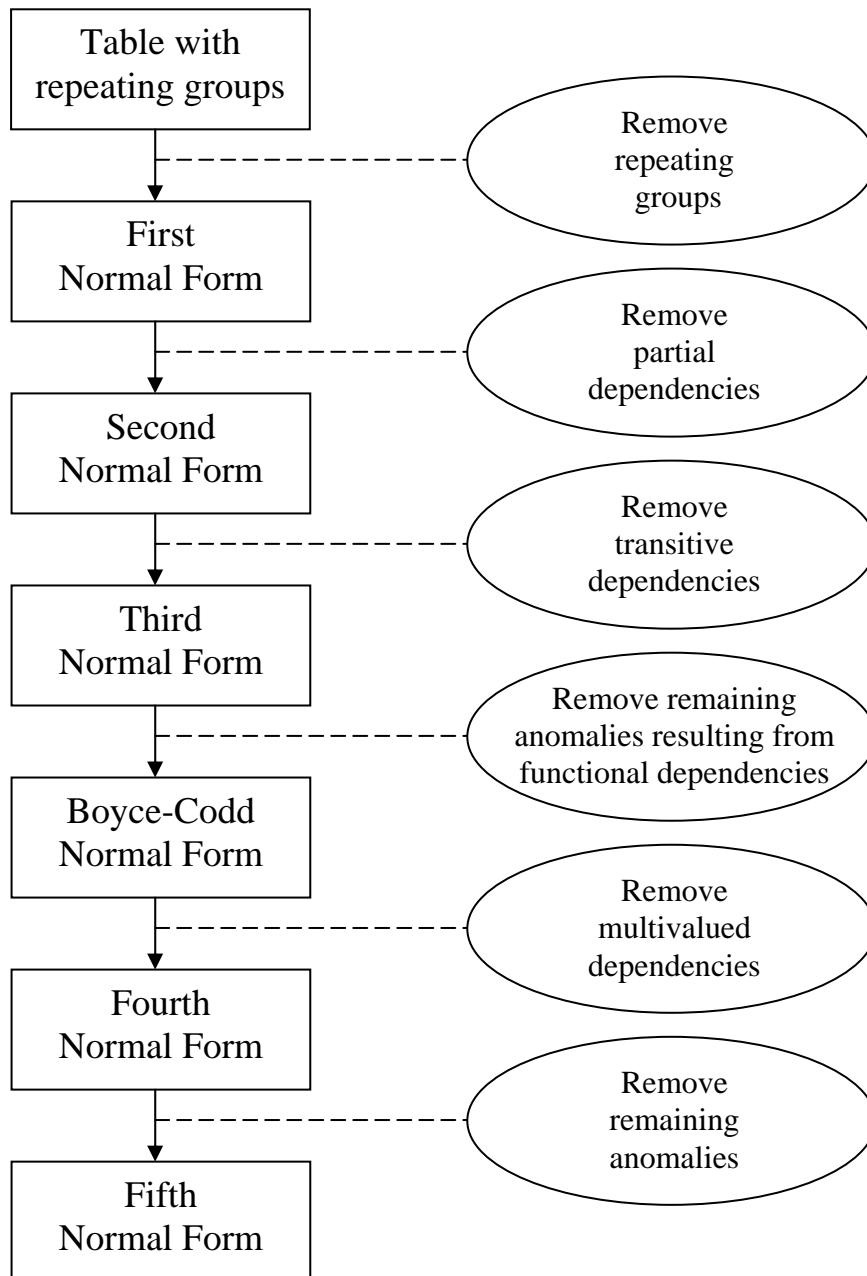
រាល់ attributes ទាំងអស់ដែលមិនបានចូលរួមបង្កើត Primary key គេហៅថា **Non-primary-key attribute** រឺ **Non-key attribute** ដែលគួរតែជាអាស្រ័យអនុគមន៍ ចំពោះ Key attribute ។



4. The Process of Normalization

ដំណើរការរបស់ normalization ជាវិធីជាផ្លូវការមួយដែលធ្វើការកំណត់ relations ដោយមានមូលដ្ឋានលើ primary key របស់ពួកគេ (រឺ candidate key ចំពោះ BCNF) និងអាស្រ័យអនុគមន៍ (functional dependency) ចំពោះ attributes របស់ពួកគេ។ វិធីនេះពាក់ព័ន្ធនឹងសេរីនៃច្បាប់ (series of rules) ដែលត្រូវបានគេប្រើប្រាស់សំរាប់ត្រួតពិនិត្យលើ relation នីមួយៗ ទើបគេអាចកំណត់កំរិតបាន។ នៅពេលដែល relation មិនទាន់បានបំពេញតាមតម្រូវការនៃ normalization ទេ relation នោះត្រូវតែបំបែកទៅជា relations មួយរឺច្រើនដែល relation នីមួយៗត្រូវតែបំពេញតម្រូវការនៃ normalization ។

Normalization ជាទូទៅដំណើរការទៅតាមសេរីជំហានដែលជំហាននីមួយៗ គេហៅថា Normal Form ។ ដំណើរការ Normalization បង្ហាញដូចរូបខាងក្រោម



(រូប 6.3)

4.1. First Normal Form (1NF)

Relations ដែលគេចាត់ទុកជា First Normal Form (1NF) កាលណា relation នោះ គ្មាន repeating groups ។ **Repeating groups** គឺជា attribute មួយ រឺច្រើនក្នុង relation ដែលបង្ហាញឡើងនូវតំលៃជាច្រើនសំរាប់ការលេចឡើងតែមួយ។ យើងធ្វើការលុប repeating groups ដោយបញ្ចូលទិន្នន័យសមរម្យទៅជា non-repeating data ទៅក្នុង column ទទេនៃ row ។

ឧទាហរណ៍ : យើងមាន Unnormalized table ដែលនឹងបំបែកទៅជា 1NF table

ដូចរូបខាងក្រោម:

Customer_Rental (Customer_No, CName, Property_No, PAddress, RentStart, RentFinish, Rent, Owner_No, OName)

Repeating Group = (Property_No, PAddress, RentStart, RentFinish, Rent, Owner_No, OName)

Customer_No	CName	Property_No	PAddress	RentStart
CR76	John Kaly	PG4	6 Lawrence St, Glasgow	1-July-94
		PG16	5 Novar Dr, Glasgow	1-Sep-96
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-92
		PG36	2 Manor Rd, Glasgow	10-Oct-94
		PG16	5 Novar Dr, Glasgow	1-Jan-96

RentFinish	Rent	Owner_No	OName
31-Aug-96	350	CO40	Tina Murphy
1-Sep-98	450	CO93	Tony Shaw
10-Jun-94	350	CO40	Tina Murphy
1-Dec-95	375	CO93	Tony Shaw
10-Aug-96	450	CO93	Tony Shaw

Customer_Rental unnormalized table. (រូប 6.4)

Customer_No	CName	Property_No	PAddress	RentStart
CR76	John Kaly	PG4	6 Lawrence St, Glasgow	1-July-94
CR76	John Kaly	PG16	5 Novar Dr, Glasgow	1-Sep-96
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-92
CR56	Aline Stewart	PG36	2 Manor Rd, Glasgow	10-Oct-94
CR56	Aline Stewart	PG16	5 Novar Dr, Glasgow	1-Jan-96



RentFinish	Rent	Owner_No	OName
31-Aug-96	350	CO40	Tina Murphy
1-Sep-98	450	CO93	Tony Shaw
10-Jun-94	350	CO40	Tina Murphy
1-Dec-95	375	CO93	Tony Shaw
10-Aug-96	450	CO93	Tony Shaw

First Normal Form (1NF) Customer_Rental relation. (រូប 6.5)

4.2. Second Normal Form (2NF)

Second Normal Form (2NF) មានមូលដ្ឋានត្រឹមត្រូវទៅលើ Concept of Full Functional Dependency ។

4.2.1 Full Functional Dependency

ឧបមាថាយើងមាន A និង B គឺជា attributes នៃ relation មួយ។ អាស្រ័យអនុគមន៍ (Functional Dependency) $A \longrightarrow B$ ជាអាស្រ័យអនុគមន៍ពេញលេញ (Full Functional Dependency) កាលណាបើយើងលុប attribute ណាមួយចេញពី A នោះ B មិនមែនជាអាស្រ័យអនុគមន៍នៃ A ទៀតទេ។ អាស្រ័យអនុគមន៍ $A \longrightarrow B$ ជាអាស្រ័យអនុគមន៍

មិនពេញលេញ (Partial Functional Dependency) កាលណាបើយើងលុប attribute ណាមួយចេញពី A នោះ B នៅតែជាអាស្រ័យអនុគមន៍នៃ A ដដែល។

ឧទាហរណ៍ : ចំពោះ Customer_Rental relation យើងកំណត់អាស្រ័យអនុគមន៍ ដូចខាងក្រោម

$$\text{Customer_No, Property_No} \longrightarrow \text{CName}$$

វាជាអាស្រ័យអនុគមន៍ត្រឹមត្រូវដែល Customer_No និង Property_No ចូលរួមគ្នាកំណត់ CName ប៉ុន្តែវាមិនមែនជាអាស្រ័យអនុគមន៍ពេញលេញទេ ពីព្រោះ CName ក៏អាចជាអាស្រ័យអនុគមន៍ចំពោះសំនុំរងនៃ (Customer_No, Property_No) គឺ Customer_No ។

4.2.2 Definition of Second Normal Form

Second Normal Form (2NF) ត្រូវបានគេប្រើជាមួយ relations ដែលមាន **Composite key** (Composite key គឺជា Primary key ដែលកើតឡើងដោយ attributes 2 រឺច្រើន)។ Relation ដែលមាន primary key កើតឡើងដោយ 1 attribute យ៉ាងហោច ណាស់ស្ថិតក្នុង 2NF ដែរ។

Relation ដែលគេចាត់ថាស្ថិតក្នុង 2NF លុះត្រាតែ relation នោះជា First Normal Form (1NF) និងគ្រប់ non-primary-key attributes ទាំងអស់សុទ្ធតែជាអាស្រ័យអនុគមន៍ពេញលេញនៃ primary key រឺ relation នោះគ្មានអាស្រ័យអនុគមន៍មិនពេញលេញ (partial functional dependency) ។

ដើម្បីបំប្លែង relation ទំរង់ 1NF ទៅ 2NF វាមានការពាក់ព័ន្ធដល់ការលុប អាស្រ័យអនុគមន៍មិនពេញលេញ (Partial functional dependencies)។ ប្រសិនបើមាន អាស្រ័យអនុគមន៍មិនពេញលេញកើតមាន យើងត្រូវលុប attribute អាស្រ័យអនុគមន៍ (functionally dependent attributes) ចេញពី relation នោះដោយយកពួកវាទៅដាក់ក្នុង relation ថ្មីជាមួយ copy of determinant ។

ឧទាហរណ៍ :

Customer_Rental (Customer_No, CName, Property_No, PAddress, RentStart, RentFinish, Rent, Owner_No, OName)

យោងតាមទិន្នន័យយើងធ្វើការកំណត់អាស្រ័យអនុគមន៍ចំពោះ Customer_Rental relation ដូចខាងក្រោម:

$$\text{Customer_No, Property_No} \longrightarrow \text{RentStart, RentFinish}$$

(Primary key)

Customer_No \longrightarrow CName

(Partial functional dependency)

Property_No \longrightarrow PAddress, Rent, Owner_No, OName

(Partial functional dependency)

Owner_No \longrightarrow OName

(Transitive dependency)

គេត្រូវធ្វើការបំបែក Customer_Rental relation ទៅជា 3 relations ដែល relations ទាំងនោះសុទ្ធតែជា 2NF

Customer (Customer_No, CName)

Property_Owner (Property_No, PAddress, Rent, Owner_No, OName)

Rental (Customer_No, Property_No, RentStart, RentFinish)

4.3. Third Normal Form (3NF)

ទោះបីជា 2NF relations មានភាពលើសទិន្នន័យតិចជាង 1NF ក៏ដោយ ក៏វានៅតែមានបញ្ហា Update anomalies កើតឡើងដដែល។ Update anomalies ក្នុង 2NF បង្កឡើងដោយអាស្រ័យអនុគមន៍ឆ្លង (Transitive dependency) ដូច្នេះយើងត្រូវលុបបំបាត់អាស្រ័យអនុគមន៍ឆ្លងដើម្បីអោយវាក្លាយទៅជា 3NF relations ។

4.3.1 Transitive Dependency

Transitive dependency (អាស្រ័យអនុគមន៍ឆ្លង) គឺជាអាស្រ័យអនុគមន៍រវាង non-primary-key attributes 2 រឺច្រើនក្នុង relation មួយ។

A, B និង C សុទ្ធតែជា attributes នៃ relation មួយដែលមាន $A \longrightarrow B$ និង $B \longrightarrow C$ ដូចនេះគេថា C ជាអាស្រ័យអនុគមន៍ឆ្លងនៃ A តាមរយៈ B ។

ឧទាហរណ៍ :

Property_No \longrightarrow Owner_No និង Owner_No \longrightarrow OName

ដូចនេះគេថា អាស្រ័យអនុគមន៍ឆ្លង Property_No \longrightarrow OName កើតមានតាមរយៈ Owner_No attribute ។

4.3.2 Definition of Third Normal Form

Relation ដែលគេចាត់ថាស្ថិតក្នុង 3NF លុះត្រាតែ relation នោះជា First Normal Form (1NF) និង Second Normal Form (2NF) ព្រមទាំងគ្មាន non-primary-key attributes ណាមួយជាអាស្រ័យអនុគមន៍ឆ្លងលើ primary key ។

ដើម្បីបំប្លែង relation ទំរង់ 2NF ទៅ 3NF វាមានការពាក់ព័ន្ធដល់ការលុប អាស្រ័យអនុគមន៍ឆ្លង (Transitive dependencies)។ ប្រសិនបើមានអាស្រ័យអនុគមន៍ឆ្លង កើតមាន យើងត្រូវលុប attribute អាស្រ័យអនុគមន៍ឆ្លង (transitively dependent attributes) ចេញពី relation នោះដោយយកពួកវាទៅដាក់ក្នុង relation ថ្មីជាមួយ copy of determinant ។

ឧទាហរណ៍ : អាស្រ័យអនុគមន៍ចំពោះ Customer, Property_Owner និង

Rental relations មានដូចខាងក្រោម

Customer Relation

Customer_No \longrightarrow CName

Property_Owner Relation

Property_No \longrightarrow PAddress, Rent, Owner_No, OName

Owner_No \longrightarrow OName

Rental Relation

Customer_No, Property_No \longrightarrow RentStart, RentFinish

ដោយសារតែ Property_Owner relation មានអាស្រ័យអនុគមន៍ឆ្លងទៅលើ

Primary key ដូចនេះគេត្រូវបំបែកវាទៅជា 2 relations

Property_for_Rent (Property_No, PAddress, Rent, Owner_No)

Owner (Owner_No, OName)

ដូចនេះ Customer_Rental relation ដែលត្រូវបំបែកជា 3NF relations

មានដូចជា

Customer (Customer_No, CName)

Rental (Customer_No, Property_No, RentStart, RentFinish)

Property_for_Rent (Property_No, PAddress, Rent, Owner_No)

Owner (Owner_No, OName)

Review Questions

1. ចូរពន្យល់អោយបានច្បាស់រូបរាងនូវពាក្យខាងក្រោម
 - (a) Normalization
 - (b) Functional Dependency
 - (c) Composite key
2. អ្វីទៅជា Update anomalies? តើវាចែកជាប៉ុន្មាន? ចូរពន្យល់អោយបានច្បាស់រូបរាង។
3. ចូរពន្យល់ពីដំណាក់កាលនីមួយៗនៃ normalization (1NF, 2NF, 3NF)?

