ជំពូកទី ៣

Relational Model

The Relational Database Management System (RDBMS) បានក្លាយទៅជា data-processing software ដែលលេចធ្លោជាងគេនាពេលបច្ចុប្បន្ននេះ ហើយតំលៃក្នុងការលក់ប្រហែលចន្លោះរវាង $8-$10 billions ក្នុងមួយឆ្នាំ និងអត្រាក្នុងការរីកលូតលាស់គឺ 25% ក្នុងមួយឆ្នាំ។ Software នេះបានក្លាយទៅជា DBMS ជំនាន់ទី 2 ដែលមានមូលដ្ឋានគ្រឹះលើ relational data model ស្នើដោយ E. F. Codd ក្នុងឆ្នាំ 1970។ នៅក្នុង relational model ទិន្នន័យទាំងអស់ត្រូវបានរក្សាទុកក្នុងទំរង់ជា relations (tables) ហើយ relation នីមួយៗតែងតែមានឈ្មោះ និងកើតឡើងពីបន្សំនៃ attributes (columns) ជាច្រើន។

1. Brief History of the Relational Model

Relational model ត្រូវបានលើកសំណើជាលើកដំបូងដោយ E. F. Codd នៅក្នុងឆ្នាំ 1970 នៅក្នុង seminal paper មួយនិយាយពី 'A relational model of data for large shared data banks' ហើយ paper នេះត្រូវបានគេចាត់ទុកជា landmark នៅក្នុង database systems។ គោលដៅនៃ relational model គឺផ្តល់នូវចំនុចដូចខាងក្រោម:

- Data independence មានកំរិតខ្ពស់ មានន័យថា application programs ត្រូវតែគ្មានការប៉ះពាល់ រឺកែប្រែ ប្រសិនបើមានការកែប្រែទៅលើ internal data representation ដូចជា changes of file organization, record orderings, and access paths។

- ផ្តល់នូវមូលដ្ឋានគ្រប់គ្រាន់សំរាប់ប្រើប្រាស់ជាមួយ data semantics, consistency, and redundancy problems។ ជាពិសេស Codd's paper បាននែនាំអោយស្គាល់នូវ **normalized relations** គឺថា relations ដែលគ្មាន repeating groups។

- ធ្វើអោយយើងមានលទ្ធភាពក្នុងការពង្រីកបន្ថែមទៅលើ set-oriented data manipulation languages។

ដោយមើលឃើញទៅលើផលប្រយោជន៍នៃ relational model ទើបមានការកសាង database ដោយប្រើប្រាស់បច្ចេកវិទ្យានេះ។ ក្នុងចំណោមនោះមាន projects 3 ធំៗគួរអោយកត់សំគាល់ ទី 1 គឺក្នុងឆ្នាំ 1970 នៅឯ IBM's San José Research Laboratory in

California មានការកកើត prototyping relational DBMS, System R។ Project នេះត្រូវបានគេកសាងឡើង ដើម្បីបង្ហាញជាភស្តុតាងនូវការប្រើប្រាស់ relational model ដោយប្រតិបត្តិទៅលើ data structures and operations របស់វា។ វាក៏បានសបង្ហាញផងដែរនូវ ប្រភពព័ត៌មានដ៏ប្រសើរអំពីការប្រតិបត្តិទៅលើ concurrency control, query optimization, transaction management, data security and integrity, recovery techniques, human factors, and user interfaces ព្រមទាំងបណ្តាលអោយការចោះពុម្ពផ្សាយទៅលើ research papers ជាច្រើន និងការអភិវឌ្ឍន៍ prototyping ដ៏ទៃទៀត។ ជាពិសេសនៅក្នុង System R project នាំអោយមានការអភិវឌ្ឍន៍:

- ការអភិវឌ្ឍន៍ Structured Query Language (pronounced ‘S-Q-L’ or sometimes ‘See-Quel’) ដែលតាំងពីពេលនោះបានក្លាយទៅជា formal International Organization of Standardization (ISO) និង *de facto* standard language of relational database។

- មានការកកើតឡើងនូវ commercial relational DBMS products ជាច្រើនក្នុង កំឡុងឆ្នាំ 1980 ឧទាហរណ៍: DB2 from IBM and ORACLE from ORACLE Corporation។

Project ទី 2 ក៏បានក្លាយជាផ្នែកមួយដ៏សំខាន់ផងដែរក្នុងការអភិវឌ្ឍន៍ relational model គឺ INGRES (INteractive GRaphics REtrieval System) project at the University of California at Berkeley ដែលដំណើរការយ៉ាងសកម្មក្នុងកំឡុងពេលជាមួយនឹង System R project។ INGRES project ពាក់ព័ន្ធនឹងការអភិវឌ្ឍន៍ prototyping RDBMS ជាមួយនឹង ការផ្តោតអារម្មណ៍ទៅលើគោលបំណងដូចគ្នានឹង System R project។ Project នេះធ្វើអោយ មានកកើត commercial products INGRES from Relational Technology Inc (now CA-OpenIngres from Computer Associates) and the Intelligent Database Machine from Briton Lee Inc។

Project ទី 3 គឺ Peterlee Relational Test Vehicle at the IBM UK Scientific Center in Peterlee ដែលមានលក្ខណៈ theoretically orientaiton ជាង System R and INGRES projects ហើយវាមានសារៈសំខាន់ក្នុងការស្រាវជ្រាវចំពោះបញ្ហាដូចជា query processing and optimization, and functional extension។

ដោយយោងទៅលើប្រជាប្រិយភាពរបស់ relational model ទើប non-relational system ជាច្រើនបច្ចុប្បន្ននេះបានបញ្ចូលលក្ខណៈ relational ទៅកាន់ model របស់ខ្លួន។

2. Terminology

Relational model មានមូលដ្ឋានគ្រឹះលើ mathematical concepts នៃ relation ដែលបង្ហាញអោយឃើញក្នុងទំរង់ជា table។ Codd ដែលមុខងារជាគ្រូបង្រៀនគណិតវិទ្យាបានប្រើប្រាស់ terminology ចេញពីគណិតវិទ្យាគឺ set theory។

2. 1. Relational Data Structure

Relation A relation is a table with columns and rows.

Relation គឺជា table ដែលផ្សំឡើងដោយ columns និង rows។

នៅក្នុង relational DBMS, users យល់ថា database ដែលគេបង្កើតឡើងដើម្បីរក្សាទុកទិន្នន័យក្នុងទំរង់ជា tables។ ទោះបីជាយ៉ាងណាក៏ដោយគួរកត់សំគាល់ដែរថា ការយល់ឃើញបែបនេះសំដៅទៅលើតែ logical structure នៃ database តែប៉ុណ្ណោះ គឺថា external និង conceptual levels នៃ ANSI-SPARC architecture។ វាមិនបានសំដៅទៅលើ physical structure នៃ database ដែលអាចអនុវត្តអោយចេញជារូបរាងតាមរយៈ storage structures ជាច្រើនបែប។

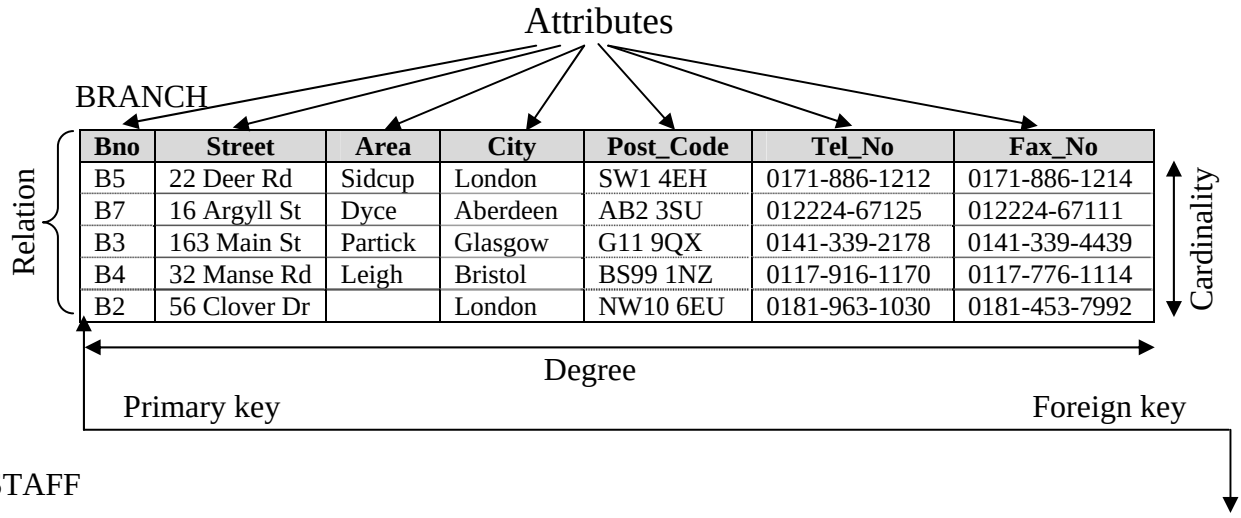
Attribute An attribute is a named column of a relation.

Attribute គឺជាឈ្មោះនៃជួរឈរក្នុង relation។

Relations ត្រូវបានគេប្រើប្រាស់សំរាប់ផ្ទុកព័ត៌មានអំពី objects បង្ហាញក្នុង database ចំពោះ relational model។ Relation បង្ហាញក្នុងទំរង់ជា two-dimensional table ដែលមាន row នៃ table សំរាប់សំគាល់ records នីមួយៗ និង column សំរាប់សំគាល់ attribute។ Attributes អាចបង្ហាញតាមលំដាប់ណាក៏បាន មានន័យថា ទោះបីជា attribute នៅមុខ រឺនៅក្រោយក៏ដោយ (ឧទាហរណ៍: Sex attribute នៅមុខ Name attribute) វានៅតែទទួលបានព័ត៌មានដូចគ្នាដដែល។

ឧទាហរណ៍: ព័ត៌មាននៅក្នុង branch office បង្ហាញតាមរយៈ Branch relation ដែលមាន attributes ដូចជា Bno (branch number), Street, Area, City, Post_Code, Tel_No និង Fax_No។ ស្រដៀងគ្នានេះដែរ ព័ត៌មានអំពី staff បង្ហាញតាមរយៈ Staff relation ដែលមាន attributes ដូចជា Sno (staff number), FName, LName, Address, Tel_No, Position, Sex, DOB (date of birth), Salary, NIN (national insurance

number) និង Bno (ជាលេខកូដរបស់ branch ដែល staff នោះស្ថិតនៅ)។ តាមរយៈរូប 3.1 ដែលបង្ហាញពី instances នៃ Branch និង Staff relations យើងសង្កេតឃើញថា column នីមួយៗសុទ្ធតែផ្ទុកតំលៃនៃ single-valued attribute ឧទាហរណ៍ Bno column ផ្ទុកតំលៃលេខកូដតែមួយចំពោះ branch ដែលបានបង្កើតរួចហើយ។



Bno	Street	Area	City	Post_Code	Tel_No	Fax_No
B5	22 Deer Rd	Sidcup	London	SW1 4EH	0171-886-1212	0171-886-1214
B7	16 Argyll St	Dyce	Aberdeen	AB2 3SU	012224-67125	012224-67111
B3	163 Main St	Partick	Glasgow	G11 9QX	0141-339-2178	0141-339-4439
B4	32 Manse Rd	Leigh	Bristol	BS99 1NZ	0117-916-1170	0117-776-1114
B2	56 Clover Dr		London	NW10 6EU	0181-963-1030	0181-453-7992

STAFF

Sno	FName	LName	Address	Tel_No	Position	Sex	DOB	Salary	NIN	Bno
SL21	John	White	19 Taylor St, Cranford, London	0171-884-5112	Manager	F	1-Oct-45	30000	WK442011B	B5
SG37	Ann	Beech	81 George St, Glasgow PA1 2JR	0141-848-3345	Snr Asst	F	10-Nov-60	12000	WL432514C	B3
SG14	David	Ford	63 Ashby St, Partick, Glasgow G11	0141-339-2177	Deputy	M	24-Mar-58	18000	WL220658D	B3
SA9	Mary	Howe	2 Elm Pl, Aberdeen AB2 3SU		Assistant	F	19-Feb-70	9000	WM532187D	B7
SG5	Susan	Brand	5 Gt Western Rd, Glasgow G12	0141-334-2001	Manager	F	3-Jun-40	24000	WK588932E	B3
SL41	Julie	Lee	28 Malvern St, Kilburn NW2	0181-554-3541	Assistant	F	13-Jun-65	9000	WA290573K	B5

(រូប 3.1)

Domain A domain is the set of allowable values for one or more attributes.

Domain គឺជាសំនុំនៃតំលៃដែលអនុញ្ញាតិអោយបញ្ចូលចំពោះ attributes 1 រឺច្រើន។ Domain បង្ហាញពីលក្ខណៈដ៏អស្ចារ្យមួយនៅក្នុង relational model ហើយគេតែងតែកំនត់ domain រាល់ attribute ទាំងអស់នៅក្នុង relational database។ Domain មួយចំនួនប្រើប្រាស់បានតែទៅលើ attribute នោះតែប៉ុណ្ណោះ តែមាន domain មួយចំនួនប្រើប្រាស់លើ attributes 2 រឺច្រើនផងដែរ។ រូប 3.2 បង្ហាញពី domain នៃ attributes មួយចំនួននៃ Branch និង Staff relations។ ទោះបីជានៅក្នុង Branch relation មាន attributes ចំនួន 7 ក៏ដោយក៏ប៉ុន្តែវាមាន 6 domains ប៉ុណ្ណោះព្រោះ attributes Tel_No និង Fax_No ទាញយកតំលៃចេញពី domain តែមួយ។

Attribute	Domain Name	Meaning	Domain Definition
Bno	BRANCH_NUMBERS	The set of all possible branch numbers	character: size 3, range B1-B99
Street	STREET_NAMES	The set of all street names in Britain	character: size 25
Area	AREA_NAMES	The set of all area names in Britain	character: size 20
City	CITY_NAMES	The set of all city names in Britain	character: size 15
Post_Code	POST_CODES	The set of all postcodes in Britain	character: size 8
Tel_No	TELFAX_NUMBERS	The set of all telephone and fax numbers in Britain	character: size 13
Fax_No	TELFAX_NUMBERS	The set of all telephone and fax numbers in Britain	character: size 13
Sex	SEX	The sex of a person	character: size 1, value M or F
DOB	DATES-OF-BIRTH	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
Salary	SALARIES	Possible values of staff salaries	monetary: 7 digits, range 6000.00-40000.00

(រូប 3.2)

Tuple A tuple is a row of a relation.

Tuple គឺជា 1 row នៃ relation។ Elements នៅក្នុង relation គឺ rows រឺ tuples ដែលរាល់ row ផ្ទុកនូវតំលៃជាច្រើនដែលតំលៃមួយសំរាប់ attribute មួយ។ Tuples អាចបង្ហាញតាមលំដាប់ណាក៏បាន ក៏ relation នៅតែទទួលបានព័ត៌មានដូចគ្នាដដែល។

Degree The degree of a relation is the number of attributes it contains.

Degree នៃ relation គឺជាចំនួន attributes ដែល relation មាន។ ដូចនេះគេថា Branch relation នៅក្នុងរូប 3.1 មាន 7 attributes រឺ degree seven នេះមានន័យថា រាល់ row នៃ relation នោះមានផ្ទុកតំលៃចំនួន 7។ Relation ដែលមាន 2 attributes គេហៅថា **binary**, 3 attributes គេហៅថា **ternary** និងច្រើនជាង 3 attributes គេហៅថា **n-ary**។

Cardinality The cardinality of a relation is the number of tuples it contains.

Cardinality នៃ relation គឺជាចំនួន tuples ដែល relation មាន។

Relational database A collection of normalized relations.

Relational database គឺជាសំនុំនៃ relations ដែលបានឆ្លងកាត់ដំណើរការនៃ normalization។ Relational database តែងតែមាន relations ជាច្រើនដែលសមស្របទៅតាមរចនាសម្ព័ន្ធ។

Alternative terminology

មាន terminology ផ្សេងទៀតក្នុងការកំនត់ terms ចំពោះ relational model។ Relation អាចហៅថា **file**, tuple អាចហៅថា **record** និង attribute អាចហៅថា **field**។

Formal terms	**Alternative 1**	**Alternative 2**
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

(រូប 3.3)

2. 2. Properties of Relations

Relation មួយតែងតែមានលក្ខណៈដូចខាងក្រោម:

- Relation មានឈ្មោះមួយដែលខុសពីឈ្មោះ relations ផ្សេងទៀតនៅក្នុង database តែមួយ។ (The relation has a name that is distinct from all other relation names.)
- Cell នីមួយៗនៃ relation ផ្ទុកតំលៃតែមួយគត់។ (Each cell of the relation contains exactly one atomic (single) value.)
- Attribute មានឈ្មោះមួយ ដែលខុសពីឈ្មោះ attributes ផ្សេងទៀតនៅក្នុង relation តែមួយ។ (Each attribute has a distinct name.)
- តំលៃបញ្ចូលទៅកាន់ attribute សុទ្ធតែទាញចេញពី domain ដូចគ្នា។ (The values of an attributes are all from the same domain.)
- រាល់ tuple នីមួយៗសុទ្ធតែខុសគ្នា។ (Each tuple is distinct; there are no duplicate tuples.)
- លំដាប់នៃ attributes គ្មានសារៈសំខាន់ទេ។ (The order of attributes has no significance.)
- តាមទ្រឹស្តី លំដាប់នៃ tuples គ្មានសារៈសំខាន់ទេ ក៏ប៉ុន្តែនៅក្នុងការអនុវត្តជាក់ស្តែង លំដាប់នៃ tuples អាចប៉ះពាល់ដល់ប្រសិទ្ធភាពក្នុងការចូលប្រើប្រាស់ tuples។ (The order of tuples has no significance, theoretically. However, in practice, the order may affect the efficiency of accessing tuples.)

2. 3. Relational Keys

យើងដឹងថា នៅក្នុង relation គ្មាន tuples ដែលមានតំលៃដូចគ្នាទេ ដូចនេះយើងតែអាចធ្វើការកំនត់អត្តសញ្ញាណទៅលើ tuple នីមួយៗនៃ relation បានដោយ attribute 1 រឺច្រើនចេញពី relation នោះ។ Relation keys ចែកជា:

Superkey An attribute or a set of attributes that uniquely identifies a tuple within a relation.

Superkey គឺជា attributes 1 រឺច្រើនដែលអាចធ្វើការកំនត់អត្តសញ្ញាណទៅលើ tuple នៃ relation។ Superkey អាចផ្តុំកន្លះ attributes លើស (additional attributes) ដែលមិនចាំបាច់ទាល់តែសោះក្នុងការកំនត់អត្តសញ្ញាណ ដូចនេះនៅពេលដែលធ្វើការកំនត់ន្លះ superkey គេតែងតែជ្រើសរើស attributes ណាដែលចាំបាច់សំរាប់កំនត់អត្តសញ្ញាណតែប៉ុណ្ណោះ។

ឧទាហរណ៏: ចំពោះ Branch relation ក្នុងរូប 3.1 យើងកំនត់បានន្លះ superkey ជាច្រើន ដូចជា:

- Bno
- Bno, Address
- Tel_No
- Fax_No
- Post_Code

Candidate key A superkey such that no proper subset is a superkey within the relation.

Candidate key គឺជា superkey ណាដែលគ្មានសំនុំរង រឺបំណែកនៃ superkey នោះអាចកំនត់ជា superkey ក្នុង relation នោះបាន។ Candidate key *K* សំរាប់ relation *R* មានលក្ខណ: 2 គឺ

- Uniqueness: រាល់ tuple ទាំងអស់នៃ R តំលៃ K អាចធ្វើការកំនត់អត្តសញ្ញាណ tuple នោះបាន។ (In each tuple of R, the values of K uniquely identify that tuple.)

- Irreducibility: គ្មានបំណែកនៃ K អាចមានលក្ខណ:តែមួយទេ។ (No proper subset of K has the uniqueness property.)

នៅក្នុង relation មួយអាចមាន candidate keys ជាច្រើន។ នៅពេលដែល key នោះកើតឡើងដោយសារបន្សំនៃ attributes ចាប់ពី 2 ឡើងទៅ គេហៅ key នោះថា **composite key**។ ចូរសង្កេតមើលរូប 3.1 យើងឃើញថា តំលៃ City អាចធ្វើការកំនត់ទៅលើ branch offices បានមួយចំនួន ដោយមូលហេតុថា London មាន branch offices ដល់ទៅ 2 ដូចនេះ attribute City មិនត្រូវបានគេជ្រើសរើសជា candidate key ទេ។ ម្យ៉ាងវិញទៀត ក្រុមហ៊ុនបាន

កំនត់លេខកូដ (Bno) រាល់ branch នីមួយៗមិនដូចគ្នាទេ ដូចនេះគេអាចកំនត់យក Bno ធ្វើជា candidate key។ ស្រដៀងគ្នាផងដែរ គេអាចជ្រើសរើស Tel_No និង Fax_No ធ្វើជា candidate keys ផងដែរ។

ពេលនេះចូរក្រឡេកមើលរូប 3.4 ខាងក្រោមនេះចំពោះ Viewing relation ដែលប្រើប្រាស់សំរាប់ផ្តុកព័ត៌មានទាក់ទងដល់ renters បានទៅមើល (view) properties។ Relation នេះមាន attributes ដូចជា renter number (Rno), property number (Pno), date of viewing (Date) និង Comment។ Rno មួយអាចធ្វើការមើលទៅលើ properties ជាច្រើនខុសៗគ្នា ហើយ Pno មួយអាចត្រូវ renters ផ្សេងៗគ្នាមើលបានដូចគ្នាដែរ ហេតុដូចនេះទាំង Rno និង Pno មិនអាចជ្រើសរើសជា candidate keys បានទេ ព្រោះវាមិនអាចកំនត់អត្តសញ្ញាណ tuple បាន។ ផ្ទុយទៅវិញ ការរួមបញ្ចូលគ្នារវាង Pno និង Rno អាចធ្វើការកំនត់អត្តសញ្ញាណបានភាគច្រើន។ ប្រសិនបើយើងគិតទៅដល់ថា renter ម្នាក់អាចមើល property ដដែលនោះលើសពីម្តងក្នុងពេលវេលាខុសគ្នា ដូច្នេះគេអាចទាញ Date បន្ថែមទៅកាន់ composite key ដែលមាន 2 attributes ស្រាប់ធ្វើអោយចុងក្រោយ candidate key មាន attributes សរុប 3 គឺ Pno, Rno និង Date។

គួរកត់សំគាល់ផងដែរថា instances មួយចំនួននៃ relation មិនត្រូវបានគេប្រើប្រាស់ដើម្បីបង្ហាញថា attributes 1 រឺច្រើនអាចកំនត់ជា candidate key បានទេ ពីព្រោះនៅពេលបច្ចុប្បន្ននេះ វាគ្មានតំលៃស្ងួនទេប៉ុន្តែនៅពេលអនាគតវាអាចមានតំលៃស្ងួនកើតមាន។ នៅពេលធ្វើការកំនត់ candidate key ទាមទារអោយយើងអោយបានច្បាស់នូវអត្ថន័យជាក់ស្តែងនៃ attributes ដែលបានចូលរួមក្នុង relation ដើម្បីអាចកំនត់តំលៃស្ងួនថា វាកើតនៅលើ attributes ណាខ្លះ។ ឧទាហរណ៍: យោងទៅលើទិន្នន័យនៃ Staff relation ដែលបង្ហាញក្នុងរូប 3.1 យើងប្រហែលជាគិតថា LName (surname) អាចធ្វើការជ្រើសរើស candidate key បានទោះបីជាយ៉ាងណាក៏ដោយ ប្រសិនបើមានសមាជិក staff ថ្មីដែលមកធ្វើការជាមួយគ្នាមាន surname White ដូចគ្នាដែរតើយើងត្រូវបញ្ចូលទិន្នន័យនោះដោយវិធីណា ប្រសិនបើគេកំនត់ LName ធ្វើជា candidate key។

ឧទាហរណ៍: ចំពោះ Branch relation ក្នុងរូប 3.1 យើងកំនត់បាននូវ candidate key ជាច្រើន ដូចជា:

- Bno
- Tel_No
- Fax_No

Primary key The candidate key that is selected to identify tuples uniquely within the relation.

Primary key គឺជា candidate key ដែលជ្រើសរើសដើម្បីធ្វើការកំនត់អត្តសញ្ញាណ tuples ក្នុង relation។ នៅពេល relation គ្មាន tuple ដែលមានតំលៃដូចគ្នា វីស្ទួនទេ ដូចនេះជាទូទៅគេតែងតែអាចកំនត់អត្តសញ្ញាណ relation នោះបាន មានន័យថា នៅក្នុង relation តែងតែមាន primary key មួយជាដាច់ខាត។ ក្នុងករណីស្មុគស្មាញបំផុត attributes ទាំងអស់អាចធ្វើការកំនត់ជា primary key ប៉ុន្តែជាធម្មតា បំណែកនៃ attributes ទាំងអស់គ្រប់គ្រាន់ក្នុងកំនត់អត្តសញ្ញាណបាន។ Candidate keys ដែលមិនបានជ្រើសរើសជា primary key គេអោយឈ្មោះថា **alternate keys**។

ឧទាហរណ៍: ចំពោះ Branch relation ក្នុងរូប 3.1 យើងធ្វើការជ្រើសរើស Bno ធ្វើជា primary key ដូចនេះ Tel_No និង Fax_No ក្លាយជា alternate keys។ ចំពោះ Viewing relation ក្នុងរូប 3.4 មាន candidate key តែមួយគត់កើតពី Bno និង Pno ដូចនេះ candidate key នីងក្លាយទៅជា primary key ដោយស្វ័យប្រវត្តិតែម្តង។

Foreign key An attribute or set of attributes within one relation that matches the candidate key of some (possibly the same) relation.

Foreign key គឺជា attributes 1 វីច្រើនក្នុង relation មួយផ្គូរជាមួយ វីមានតំលៃចង្អុលទៅកាន់ candidate key នៃ relation ដូចគ្នា វីផ្សេងទៀត។ នៅពេលដែល attribute បង្ហាញក្នុង relation លើសពីមួយ វាតែងតែបង្ហាញពី relationship រវាង tuples នៃ relations ទាំង 2 នោះ។

ឧទាហរណ៍: ការបញ្ចូល attribute Bno ទៅកាន់ relations ទាំង 2 គឺ Branch និង Staff relations សំរាប់ធ្វើការបង្ហាញពីការតភ្ជាប់រវាង branch ទៅកាន់ព័ត៌មានលំអិតនៃ staff ដែលធ្វើការនៅតាម branch នីមួយៗនោះ។ នៅក្នុង Brach relation, Bno មានតួនាទីជា primary key តែក្នុង Staff relation, Sno មានតួនាទីជា primary key និង Bno មានតួនាទីជា foreign key ប្រើប្រាស់សំរាប់ភ្ជាប់ staff ទៅកាន់ branch office ដែលពួកគេធ្វើការ។ យើងអាចនិយាយបានថា attribute Bno នៅក្នុង Staff relation ចង្អុលទៅកាន់តំលៃ primary key attribute នៅក្នុង home relation Branch ហើយ attributes ទាំងនេះដើរតួនាទីយ៉ាងសំខាន់សំរាប់ដំណើរការ data manipulation។

2. 4. Representing Relational Database Schemas

Relation schema ជាឈ្មោះ relation ដែលមានសំនុំនៃ attributes ទាំងអស់ និង ឈ្មោះ domain ជាដៃគូរបស់វានៅពីក្រោយ។

ឧទាហរណ៍: Relation schema ចំពោះ Branch relation

Branch(Bno:BRANCH_NUMBERS, Street:STREET_NAMES, Area:AREA_NAMES, City:CITY_NAMES, Post_Code:POST_CODES, Tel_No:TELFAX_NUMBERS, Fax_No:TELFAX_NUMBERS)

Relational database ផ្ទុកនូវបណ្តុំនៃ relations ជាច្រើន។ Relation schemas ចំពោះ rental database យើងមានដូចជា:

Branch	(Bno, Street, Area, City, Post_Code, Tel_No, Fax_No)
Staff	(Sno, FName, LName, Address, Tel_No, Position, Sex, DOB, Salary, NIN, Bno)
Property_for_Rent	(Pno, Street, Area, City, Post_Code, Types, Rooms, Rent, Ono, Sno, Bno)
Renter	(Rno, FName, LName, Address, Tel_No, Pref_Type, Max_Rent, Bno)
Owner	(Ono, FName, LName, Address, Tel_No)
Viewing	(Rno, Pno, Date, Comment)

Conceptual model or *conceptual schema* គឺជាសំនុំនៃ schemas បែបនេះនៅក្នុង database។ រូបខាងក្រោមបង្ហាញពី instances នៃ database នេះ

BRANCH

Bno	Street	Area	City	Post_Code	Tel_No	Fax_No
B5	22 Deer Rd	Sidcup	London	SW1 4EH	0171-886-1212	0171-886-1214
B7	16 Argyll St	Dyce	Aberdeen	AB2 3SU	012224-67125	012224-67111
B3	163 Main St	Partick	Glasgow	G11 9QX	0141-339-2178	0141-339-4439
B4	32 Manse Rd	Leigh	Bristol	BS99 1NZ	0117-916-1170	0117-776-1114
B2	56 Clover Dr		London	NW10 6EU	0181-963-1030	0181-453-7992

PROPERTY_FOR_RENT

Pno	Street	Area	City	Post_Code	Type	Rooms	Rent	Ono	Sno	Bno
PA14	16 Holhead	Dee	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B7
PL94	6 Argyll St	Kilburn	London	NW2	Flat	4	400	CO87	SL41	B5
PG4	6 Lawrence St	Partick	Glasgow	G11 9QX	Flat	3	350	CO40	SG14	B3
PG36	2 Manor Rd		Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B3
PG21	18 Dale Rd	Hyndland	Glasgow	G12	House	5	600	CO87	SG37	B3
PG16	5 Novar Dr	Hyndland	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B3

STAFF

Sno	FName	LName	Address	Tel_No	Position	Sex	DOB	Salary	NIN	Bno
SL21	John	White	19 Taylor St, Cranford, London	0171-884-5112	Manager	F	1-Oct-45	30000	WK442011B	B5
SG37	Ann	Beech	81 George St, Glasgow PA1 2JR	0141-848-3345	Snr Asst	F	10-Nov-60	12000	WL432514C	B3
SG14	David	Ford	63 Ashby St, Partick, Glasgow G11	0141-339-2177	Deputy	M	24-Mar-58	18000	WL220658D	B3
SA9	Mary	Howe	2 Elm Pl, Aberdeen AB2 3SU		Assistant	F	19-Feb-70	9000	WM532187D	B7
SG5	Susan	Brand	5 Gt Western Rd, Glasgow G12	0141-334-2001	Manager	F	3-Jun-40	24000	WK588932E	B3
SL41	Julie	Lee	28 Malvern St, Kilburn NW2	0181-554-3541	Assistant	F	13-Jun-65	9000	WA290573K	B5

RENTER

Rno	FName	LName	Address	Tel_No	Pref_Type	Max_Rent	Bno
CR76	John	Kary	56 High St, Putney, London SW1 4EH	0171-774-5632	Flat	425	B5
CR56	Aline	Stewart	64 Fern Dr, Pollock, Glasgow G42 0BL	0141-848-1825	Flat	350	B3
CR74	Mike	Ritchie	18 Tain St, Gourock PA1G 1YQ	01475-392178	House	750	B3
CR62	Mary	Tregear	5 Tarbot Rd, Kildary, Aberdeen AB9 3ST	01224-196720	Flat	600	B7

OWNER

Ono	FName	LName	Address	Tel_No
CO46	Joe	Keogh	2 Fergus Dr, Banchory, Aberdeen AB2 7SX	01224-861212
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419
CO40	Tina	Murphy	63 Well St, Shawlands, Glasgow G42	0141-943-1728
CO93	Tony	Shaw	12 Park Pl, Hillhead, Glasgow G4 0QR	0141-225-7026

VIEWING

Rno	Pno	Date	Comment
CR56	PA14	24-May-98	too small
CR76	PG4	20-Apr-98	too remote
CR56	PG4	26-May-98	
CR62	PA14	14-May-98	no dining room
CR56	PG36	28-Apr-98	

(រូប 3.4)

3. Relational Integrity

យើងដឹងហើយថា ដើម្បីបង្កើត database យើងត្រូវតែធ្វើការកំនត់លក្ខខណ្ឌដើម្បីធានាអោយបានទិន្នន័យទាំងអស់ត្រឹមត្រូវ។ នៅក្នុងចំនុចនេះ យើងនឹងនិយាយពី relational integrity rules ដែលប្រើប្រាស់ក្នុងការកំនត់ទៅលើ database ។

រាល់ attributes ទាំងអស់សុទ្ធតែមាន domain រៀងៗខ្លួនរបស់វា ដូចនេះ constraints នេះគេហៅថា **domain constraints** (domain constraints គឺជាការដាក់កំរិត (restriction) ទៅលើសំនុំតំលៃដែលអនុញ្ញាតិអោយបញ្ចូលទៅកាន់ attributes នៃ relations) ។

លើសពីនេះទៀត មាន integrity rules 2 សំខាន់ផ្សេងទៀត ដែលប្រើប្រាស់សំរាប់កំនត់លក្ខខណ្ឌ (constraints) រឺការដាក់កំរិត (restrictions) ទៅលើ instances ទាំងអស់នៃ database។ Integrity rules ទាំង 2 នោះគឺ **entity integrity** និង **referential integrity** ប៉ុន្តែមុននឹងនិយាយដល់ integrity rules នេះ យើងចាំបាច់គួរតែយល់ពី Null ជាមុនសិន។

3. 1. Nulls

Null Represents a value for an attributes that is currently unknown or is not applicable for this tuple.

Null បង្ហាញពីតំលៃសំរាប់ attributes ដែលបច្ចុប្បន្នមិនដឹង រឺមិនអាចអនុវត្តបានចំពោះ tuple នោះ។ Null ច្រើនបង្ហាញថា តំលៃលើ attributes នោះយើងមិនដឹងថាបញ្ចូលស្អី រឺមានន័យថាតំលៃលើ attributes នោះមិនទាន់បានបញ្ចូលនៅឡើយទេ។ Null មិនដូចទៅនឹងតំលៃលេខ 0 រឺ តួរអក្សរមានដកឃ្លា (text string filled with spaces)។ 0 និងដកឃ្លាបង្ហាញពីតំលៃ ប៉ុន្តែ null បង្ហាញពីអវត្តមានតំលៃ (zeros and spaces are values, but a null represents the absence of a value)។ ហេតុដូចនេះ null ត្រូវបានគេយល់ថា វាខុសពីតំលៃដ៏ទៃទៀត តួយ៉ាងដូចជា null ខុសពី ‘null values’ ។

ឧទាហរណ៍: នៅក្នុង Viewing relation បង្ហាញតាមរូប 3.4 យើងឃើញថា Comment attribute ប្រហែលជាមិនទាន់កំនត់ រឺបញ្ចូលទិន្នន័យរហូតទាល់តែ renter បានទៅមើល property ហើយបញ្ចូល Comment របស់គាត់ទៅលើ property នោះទៅកាន់ភ្នាក់ងារដែលរង់ចាំ។ ប្រសិនបើគ្មាន null ទេយើងត្រូវតែបញ្ចូលទិន្នន័យមិនត្រឹមត្រូវ (false data) ទៅកាន់ attribute នោះ រឺបន្ថែម attributes ដែលគ្មានន័យទៅកាន់ database។ នៅក្នុងឧទាហរណ៍របស់យើង យើងសាកល្បងបង្ហាញ null comment រយៈការបញ្ចូលតំលៃ -1។ ប៉ុន្តែមានវិធីម្យ៉ាងទៀតគឺយើងធ្វើការបន្ថែម attribute មួយទៀតឈ្មោះ ‘Has Comment Been Supplied?’ ទៅកាន់ Viewing relation ដែលផ្ទុកតំលៃតែ 2 គត់គឺ Y (Yes) ប្រសិនបើបានបំពេញ comment និង N (No) ប្រសិនបើមិនទាន់បានបំពេញ comment។ វិធីទាំង 2 ខាងលើនេះអាចធ្វើអោយ user មានការភាន់ច្រឡំបាន។

3. 2. Entity Integrity

Entity Integrity In a base relation, no attribute of primary key can be null.

Entity integrity គឺជាការដាក់កំរិតថា គ្មាន attribute នៃ primary key អាចផ្តុកតំលៃ null បានទេ។ **Base relation** គឺជា relation ដែលត្រូវគ្នាជាមួយ entity នៅក្នុង conceptual schema។ តាមរយៈនិយមន័យខាងលើ primary key តែងតែកើតឡើងពីបន្សំរវាង attributes ដែលមានចំនួនតិចជាងគេ ហើយប្រើប្រាស់ធ្វើការកំនត់អត្តសញ្ញាណ tuples ហេតុដូចនេះ មានន័យថាគ្មានផ្នែកណានៃ primary key មានលទ្ធភាពគ្រប់គ្រាន់ក្នុងការកំនត់អត្តសញ្ញាណបានទេ ទៅលើ tuples។ ប្រសិនបើយើងអនុញ្ញាតិអោយបញ្ចូលតំលៃ null ទៅកាន់ផ្នែកណមួយនៃ primary key នោះវាល្មើសទៅនឹងនិយមន័យរបស់ primary key។ ឧទាហរណ៍: Bno គឺជា primary key នៃ Branch relation ដូចនេះយើងមិនអាចមានលទ្ធភាពបញ្ចូល nulls ទៅកាន់ attribute នោះបានជាដាច់ខាត។

ប្រសិនបើយើងត្រួតពិនិត្យអោយកាន់តែម៉ត់ចត់ទៅលើ rules នេះទៀត យើងនឹងជួបប្រទះចំនុច 2 គួរកត់សំគាល់គឺ ទី ១ ហេតុអ្វីបានជាកំនត់ rule នេះទៅលើតែ primary key តែមិនកំនត់ទៅលើ alternate keys ផង? (First, why does the rule apply only to primary keys and not alternate keys as well?) ទី ២ ហេតុអ្វីបានជា rule នេះដាក់កំរិតទៅលើតែ base relation? (Secondly, why is the rule restricted to base relation?) ដើម្បីឆ្លើយតបទៅនឹងចំនុចខាងលើ សូមលើកយកទិន្នន័យក្នុង Staff relation ក្នុងរូប 3.4 មកធ្វើការបកស្រាយ។ ឧបមាថាយើងមាន query មួយសំរាប់បង្ហាញលេខទូរស័ព្ទ staffs ទាំងអស់ ហើយ query នេះនឹងបង្កើត unary relation ផ្តុកនូវ Tel_No attribute។ តាមរយៈចំនុចទីមួយប្រសិនបើយើងកំនត់ entity rule ទៅលើ alternate key Tel_No ផងដែរនោះមានន័យថា តំលៃលើ Tel_No មិនអាចផ្តុកតំលៃ null បានទេ។ ចំពោះ relation ដែលកើតចេញពីដំណើរការ query នេះមិនមែនជា base relation ទេ ដូចនេះវាអនុញ្ញាតិអោយមាន null ទៅលើ primary key។

3. 3. Referential Integrity

Integrity rule ទី 2 នេះកំនត់ទៅលើ foreign keys វិញ។

Referential integrity If a foreign key exists in a relation, either the foreign key values must match a candidate key value of some tuple in its home relation or foreign key value must be wholly null.

Referential integrity កំនត់ថា ប្រសិនបើ foreign key មាននៅក្នុង relation មួយ នោះតំលៃ foreign key ត្រូវតែដូច វីមាននៅក្នុងតំលៃ candidate key នៃ tuple មួយចំនួនក្នុង home relation វីតំលៃ foreign key ត្រូវតែ null។

ឧទាហរណ៍: Bno នៅក្នុង Staff relation គឺជា foreign key ដែលចង្អុលតំលៃទៅកាន់ Bno attribute នៅក្នុង home relation Branch ដូច្នេះវាមិនអាចមានលទ្ធភាពក្នុងការបញ្ចូល staff ថ្មីដែលធ្វើការនៅក្នុង branch number B25 បានទេ ទោះបីជាយ៉ាងណាក៏ដោយ យើងគួរតែមានលទ្ធភាពបញ្ចូល staff ថ្មីដែលមិនទាន់កំនត់អោយទៅធ្វើការនៅ branch ណាមួយបានដោយបញ្ចូលតំលៃ null ទៅកាន់ Bno attribute ក្នុង Staff relation។

3. 4. Enterprise Constraints

Enterprise constraints Additional rules specified by the users or database administrator of a database.

Enterprise constraints គឺជាកំនត់ rules បន្ថែមទៀតដោយអ្នកប្រើប្រាស់ វី database administrator។ Users ជាទូទៅមានលទ្ធភាពក្នុងការកំនត់ constraints បន្ថែមទៀតដែលតំរូវអោយទិន្នន័យត្រូវតែគោរពតាម។

ឧទាហរណ៍: ប្រសិនបើគេកំនត់ថា branch នីមួយៗមានសមាជិកច្រើនបំផុត 20 នាក់ ហើយរំពឹងថា DBMS នឹងជួយប្រតិបត្តិវា។ ក្នុងករណីនេះ វាមិនគួរមានលទ្ធភាពបញ្ចូលសមាជិកថ្មីទៅកាន់ branch ដែលមានសមាជិក 20 នាក់ទេ គួរអោយសោកស្តាយកំរិតក្នុងការទ្រទ្រង់ relational integrity ខុសប្លែកគ្នាពីប្រព័ន្ធមួយទៅប្រព័ន្ធមួយទៀត។

4. Views

នៅក្នុង three-level ANSI-SPARCE architecture បង្ហាញនៅជំពូកមុន យើងបានពណីនាថា external view ជារចនាសម្ព័ន្ធនៃ database ដែលវាលេចចេញទៅកាន់ user ណាមួយ។ នៅក្នុង relational model ពាក្យថា ‘view’ មានអត្ថន័យផ្សេង គឺជា **virtual relation** : relation ដែលមិនកើតមានពិតប្រាកដទេ តែវាកើតដោយស្វ័យប្រវត្តចេញពី base relation មួយ វីច្រើន (a relation that does not actually exist in its own right, but is dynamically derived from one or more other base relation.)។ គេបង្កើត view ដោយជំណើការ operations ដូចជា relational algebra selection, projection, join operations, or other calculations ទៅលើតំលៃមានស្រាប់ក្នុង base relations។

4. 1. Terminology

Base relation A name relation corresponding to an entity in the conceptual schema, whose tuples are physically stored in the database.

Base relation គឺជា relation ដែលឆ្លើយតបទៅនឹង entity ក្នុង conceptual schema ដែល tuples ត្រូវបានរក្សាទុកក្នុង database ។

View A view is the dynamic result of one or more relational operation operating on the base relations to produce another relation. A view is a **virtual relation** that does not actually exist in the database but it produced upon request by a particular user, at the time of request.

View គឺជាលទ្ធផលនៃដំណើរការ relational operations 1 ឬច្រើនទៅលើ base relations ដើម្បីបង្កើត relation ថ្មី។ View គឺជា virtual relation ដែលមិនកើតមានពិតប្រាកដនៅក្នុង database ប៉ុន្តែគេបង្កើតទៅតាមសំណើសុំរបស់ user ណាមួយកំឡុងពេលសំណើសុំ។ View អាចធ្វើការ manipulate ដូចជា base relation ដែរ ប៉ុន្តែវាមិនបានធ្វើការរក្សាទុកទិន្នន័យទៅកាន់ storage នៃ database ទេហើយ Operations ណាដែលធ្វើទៅលើ view ត្រូវបានបំលែងទៅជា operations ដំណើរការលើ relation វិញ។ View មានលក្ខណៈ **dynamic** មានន័យថា ការកែប្រែទិន្នន័យទៅលើ base relation ក៏មានឥទ្ធិពលទៅលើ view ផងដែរ គឺទិន្នន័យនៅពេលបង្ហាញនៅ view ក៏កែប្រែតាមដែរ ម្យ៉ាងវិញទៀតនៅពេលដែល user ធ្វើការកែប្រែទិន្នន័យទៅលើ view ពេលនោះ base relation ក៏កែប្រែតាមដែរ។

4. 2. Purpose of Views

គេប្រើប្រាស់ views ចំពោះហេតុផលមួយចំនួនដូចខាងក្រោម

- វាផ្តល់នូវ powerful and flexible mechanism ដោយលាក់បំពាំងផ្នែកមួយចំនួននៃ database ពី user ដូចនេះ user មិនបានដឹងថា តើ attributes ណាខ្លះដែលមានក្នុង database ឬតើ tuples ណាខ្លះដែលមិនបានបង្ហាញក្នុង view ។

- វាអនុញ្ញាតិអោយ user ធ្វើការប្រើប្រាស់ (access) ទិន្នន័យក្នុងទំរង់មួយដែលសមរម្យចំពោះតំរូវការរបស់ពួកគេ ដូចនេះទិន្នន័យដូចគ្នា តែបង្ហាញក្នុងទំរង់ខុសៗគ្នាទៅកាន់ users ផ្សេងៗគ្នាក្នុងពេលតែមួយ។

- វាអាចធ្វើអោយ complex operations ទៅលើ base relation មានលក្ខណៈងាយស្រួល។ ឧទាហរណ៍: ប្រសិនបើយើងបង្កើត view សំរាប់តភ្ជាប់ (join) រវាង relations 2

នោះ user អាចដំណើរការ simple unary operations of selection and projection ទៅលើ view នោះប្រសិនបើ user មានបំណងធ្វើការជាមួយ 2 relations។

4. 3. Updating Views

រាល់ការកែប្រែទៅលើ view ធ្វើអោយ base relation កែប្រែតាមដែរ។ ទោះបីជាយ៉ាងណាក៏ដោយ ក៏មានការដាក់កំរិតទៅលើប្រភេទនៃការកែប្រែដែលប្រើប្រាស់ទៅលើ views។ យើងធ្វើការសង្ខេបលក្ខខណ្ឌខាងក្រោមចំពោះប្រព័ន្ធភាគច្រើនកំនត់ថា ការកែប្រែអាចធ្វើទៅលើ view បានដែរឬទេ:

- យើងអាចធ្វើការកែប្រែទៅលើ views បានប្រសិនបើ views នោះជា simple query ពាក់ព័ន្ធនឹង base relation តែមួយគត់។
- យើងមិនអាចធ្វើការកែប្រែទៅលើ views បានប្រសិនបើ views នោះពាក់ព័ន្ធជាមួយ base relations ច្រើន។
- យើងមិនអាចធ្វើការកែប្រែទៅលើ views បានប្រសិនបើ views នោះពាក់ព័ន្ធ aggregating or grouping operations។

Review Questions

1. ចូរពន្យល់អោយបានក្បោះក្បាយនូវពាក្យខាងក្រោម
   (a) relation
   (b) attribute
   (c) domain
   (d) degree
   (e) cardinality
   (f) foreign key
2. ចូរពិពណ៌នាពីភាពខុសប្លែកគ្នារវាង candidate keys និង primary key?
3. អ្វីទៅជា view? តើ view ខុសពី relation ត្រង់ចំនុចណា?
4. ចូរពន្យល់ពីជំហានក្នុងការបំលែង EER diagrams ទៅជា relations?