Ch-4

NORMALIZATION
AND
RELATIONAL ALGEBRA

Normalization

- After converting the ER diagram in to table forms, the next phase is implementing the process of normalization,
  - which is a collection of rules each table should satisfy.
- Normalization is a series of steps followed to obtain a database design that allows for consistent storage and efficient access of data in a relational database.
- These steps reduce data redundancy and the risk of data becoming inconsistent.

…Normalization

- One of the best ways to determine what information should be stored in a database is
  - to clarify what questions will be asked of it and
  - what data would be included in the answers
  - identifying the logical associations between data items and
  - designing a database that will represent such associations but without suffering
    1. **Insertion Anomalies**
    2. **Deletion Anomalies**
    3. **Modification Anomalies**
- Thus, the purpose of normalization is to reduce the chances for anomalies to occur in a database.

Example of Unnormalized database table

EmpID	*FName*	*LName*	*SkillID*	*Skill*	*SkillType*	*School*	*SchoolAdd*	*Skill Level*
12	Abebe	Mekuria	2	SQL	Database	AAU	Sidist_Kilo	5
16	Lemma	Alemu	5	C++	Programming	Unity	Gerji	6
28	Chane	Kebede	2	SQL	Database	AAU	Sidist_Kilo	10
25	Abera	Taye	6	VB6	Programming	Helico	Piazza	8
65	Almaz	Belay	2	SQL	Database	Helico	Piazza	9
24	Dereje	Tamiru	8	Oracle	Database	Unity	Gerji	5
51	Selam	Belay	4	Prolog	Programming	Jimma	Jimma City	8
94	Alem	Kebede	3	Cisco	Networking	AAU	Sidist_Kilo	7
18	Girma	Dereje	1	IP	Programming	Jimma	Jimma City	4
13	Yared	Gizaw	7	Java	Programming	AAU	Sidist_Kilo	6

Anomalies

- **Deletion Anomalies**
- If employee with ***ID 16*** is deleted then
  - every information about skill ***C++*** and
  - the type of skill is deleted from the database.
- Then we will not have any information about ***C++*** and its skill type.
- **Insertion Anomalies**
- What if we have a new employee with a skill ***Pascal***?
  - We cannot decide whether ***Pascal*** is allowed as a value for skill and
  - we have no clue about the ***type of skill*** that ***Pascal*** should be categorized as.
- **Modification Anomalies**
- if the address for ***Helico*** is changed from ***Piazza*** to ***Mexico***?
  - We need to look for every occurrence of ***Helico School_Add*** from ***Piazza*** to ***Mexico***,
    - which is prone to error.

Functional Dependency (FD)

- Before moving to the definition and application of normalization,
  - it is important to have an understanding of "functional dependency."

Data Dependency

- The logical associations between data items that point the database designer in the direction of a good database design are referred to as
  - determinant or dependent relationships
    - Two data items A and B are said to be in a determinant or dependent relationship if certain values of data item B always appears with certain values of data item A

Functional Dependency (FD)

- if the existence of something, call it A, implies that B must exist and have a certain value, then
  - we say that ***"B is functionally dependent on A."***
- We also often express this idea by saying that
  - "A determines B," or
  - "B is a function of A," or
  - "A functionally governs B.“
- We ca express this as follows
  - "If A, then B."

Functional Dependency (FD)

- The notation is: A→B which is read as;
  - B is functionally dependent on A
- In general, a ***functional dependency*** is a relationship among attributes.
  - In relational databases, we can have a determinant that governs one other attribute or several other attributes.

NB: FDs are derived from the real-world constraints on the attributes.

Functional dependency example

Example

Dinner	*Type of Wine*
Meat	Red
Fish	White
Cheese	Rose

Since the type of ***Wine*** served depends on the type of ***Dinner***, we say ***Wine*** is functionally dependent on ***Dinner***.
Dinner → Wine

Dinner	*Type of Wine*	*Type of Fork*
Meat	Red	Meat fork
Fish	White	Fish fork
Cheese	Rose	Cheese fork

- Since both ***Wine*** type and ***Fork*** type are determined by the ***Dinner*** type, we say ***Wine*** is functionally dependent on ***Dinner*** and ***Fork*** is functionally dependent on ***Dinner***.

  Dinner → Wine

  Dinner → Fork

Types of dependency

- ***Partial Dependency***
- If an attribute which is not a member of the primary key is dependent on some part of the primary key (if we have composite primary key) then
  - that attribute is partially functionally dependent on the primary key.
- Let {A,B} is the Primary Key and C is non key attribute.
  - Then if **{A, B} →C and B→C or A→C**
    - Then C is partially functionally dependent on {A, B}

Types of dependency

- ***Full Dependency***
- If an attribute which is not a member of the primary key is not dependent on some part of the primary key but the whole key (if we have composite primary key) then
  - that attribute is fully functionally dependent on the primary key.
- Let {A, B} is the Primary Key and C is a non key attribute
  - Then if **{A, B} →C and B→C and A→C**
    - Then C Fully functionally dependent on {A, B}

Types of dependency

- ***Transitive Dependency***
- "If A implies B, and
  - if also B implies C, then
    - A implies C."

Example:

- ***If Mr X is a Human, and if every Human is an Animal, then Mr X must be an Animal.***
- Generalized way of describing transitive dependency is that:

If A functionally governs B, AND

If B functionally governs C

THEN A functionally governs C

- Provided that neither C nor B determines A i.e. (B /→ A and C /→ A)

Types of dependency

In the normal notation:

- **{(A→ B)** AND **(B→ C)} ==> A→C** provided that
  - **B /→ A** and
  - **C /→ A**

Steps of Normalization

- We have various levels or steps in normalization called Normal Forms.
  - as we move from one lower level Normal Form to the higher
    - The level of complexity, strength of the rule and decomposition increases.
  - A table in a relational database is said to be in a certain normal form if it satisfies certain constraints.

Steps in normalization

- **Un-Normalized Form:**
  - Identify all data elements
- **First Normal Form:**
  - Find **the key** with which you can find **all** data
- **Second Normal Form:**
  - Remove part-key dependencies.
  - Make all data dependent on **the whole key**.
- **Third Normal Form**
  - Remove non-key dependencies.
  - Make all data dependent on **nothing but the key**.
- For most practical purposes, databases are considered normalized if they adhere to third normal form.

First Normal Form (1NF)

- Definition: a table (relation) is in 1NF

If

  - There are no duplicated rows in the table. Unique identifier
  - Each cell is single-valued (i.e., there are no repeating groups).
  - Entries in a column (attribute, field) are of the same kind.

First Normal Form (1NF)

- We have two ways of achieving atomicity:
  - Putting each repeating group into a separate table and connecting them with a ***primary key-foreign key*** relationship or
  - Moving these repeating groups to a new row by repeating the common attributes.
    - If so then find the key with which you can find all data

First Normal Form (1NF)

UNNORMALIZED

EmpID	*FirstName*	*LastName*	*Skill*	*SkillType*	*School*	*SchoolAdd*	*SkillLevel*
12	Abebe	Mekuria	SQL, VB6	Database, Programming	AAU, Helico	Sidist_Kilo Piazza	5 8
16	Lemma	Alemu	C++ IP	Programming Programming	Unity Jimma	Gerji Jimma City	6 4
28	Chane	Kebede	SQL	Database	AAU	Sidist_Kilo	10
65	Almaz	Belay	SQL Prolog Java	Database Programming Programming	Helico Jimma AAU	Piazza Jimma City Sidist_Kilo	9 8 6
24	Dereje	Tamiru	Oracle	Database	Unity	Gerji	5
94	Alem	Kebede	Cisco	Networking	AAU	Sidist_Kilo	7

First Normal Form (1NF)

- In 1NF:
  - Remove all repeating groups.
  - Distribute the multi-valued attributes into different rows and
    - identify a unique identifier

EmpID	FirstName	LastName	SkillID	Skill	SkillType	School	SchoolAdd	SkillLevel
12	Abebe	Mekuria	1	SQL	Database	AAU	Sidist_Kilo	5
12	Abebe	Mekuria	3	VB6	Programming	Helico	Piazza	8
16	Lemma	Alemu	2	C++	Programming	Unity	Gerji	6
16	Lemma	Alemu	7	IP	Programming	Jimma	Jimma City	4
28	Chane	Kebede	1	SQL	Database	AAU	Sidist_Kilo	10
65	Almaz	Belay	1	SQL	Database	Helico	Piazza	9
65	Almaz	Belay	5	Prolog	Programming	Jimma	Jimma City	8
65	Almaz	Belay	8	Java	Programming	AAU	Sidist_Kilo	6
24	Dereje	Tamiru	4	Oracle	Database	Unity	Gerji	5
94	Alem	Kebede	6	Cisco	Networking	AAU	Sidist_Kilo	7

Second Normal Form (2NF)

- ***Second Normal form 2NF***
  - No partial dependency of a non-key attribute on part of the primary key.
  - This will result in a set of relations with a level of Second Normal Form.
  - Any table that is in 1NF and has a single-attribute (i.e., a non-composite) primary key is automatically in 2NF.
- **Definition: a table (relation) is in 2NF**

  If

  It is in 1NF and

  If all non-key attributes are dependent on the entire primary key. i.e. no partial dependency.

Second Normal Form (2NF)

EMP_PROJ

EmpID	EmpName	ProjNo	ProjName	ProjLoc	ProjFund	ProjMangID	Incentive

EMP_PROJ rearranged

EmpID	ProjNo	EmpName	ProjName	ProjLoc	ProjFund	ProjMangID	Incentive

- Business rule: Whenever an employee participates in a project,
  - he/she will be entitled for an incentive.

- since we don't have any repeating groups or attributes with multi-valued property.
  - This schema is in its 1NF

Second Normal Form (2NF)

- To convert it to a 2NF
  - we need to remove all partial dependencies of non-key attributes on part of the primary key.
    - ***{EmpID, ProjNo}→EmpName, ProjName, ProjLoc, ProjFund, ProjMangID, Incentive***
- But in addition to this we have the following dependencies
  - ***FD1: {EmpID} → EmpName***
  - ***FD2: {ProjNo} → ProjName, ProjLoc, ProjFund, ProjMangID***
  - ***FD3: {EmpID, ProjNo} → Incentive***
- As we can see,
  - some non-key attributes are partially dependent on some part of the primary key.
    - This can be witnessed by analyzing FD1 and FD2.

Second Normal Form (2NF)

- Thus, each Functional Dependencies, with their dependent attributes should be moved to a new relation where the Determinant will be the Primary Key for each.

EMPLOYEE

EmpID	*EmpName*

PROJECT

ProjNo	*ProjName*	*ProjLoc*	*ProjFund*	*ProjMangID*

EMP_PROJ

EmpID	*ProjNo*	*Incentive*

Third Normal Form (3NF)

- ***Third Normal Form (3NF)***
- Eliminate Columns Dependent on another non-Primary Key
- **Definition: a Table (Relation) is in 3NF**

  If

  It is in 2NF and

  There are no transitive dependencies between a primary key and non-primary key attributes.
- Example for (3NF)
  - Assumption: Students of same batch (same year) live in one building or dormitory

Third Normal Form (3NF)

Student_ID	StudFirstName	StudLasstName	Dept.	Year	Dormitory
ETS0360/13	Abel	Bekele	CS	1	201
ETS0365/12	Yonas	Tamru	SW	2	202
ETS0371/11	Lemlem	Kassa	CS	3	203
ETS0374/13	Fitsum	Bekele	SW	1	201
ETS0375/11	Aster	Alemu	SW	3	203

- This schema is in its 2NF.
- Let's take ***StudID***, ***Year*** and ***Dormitary*** and see the dependencies.
  StudID → ***Year*** AND
  Year → ***Dormitary*** And

Year cannot determine ***StudID*** and
Dormitary cannot determine ***StudID*** Then transitively
StudID → ***Dormitary***

- To convert it to a 3NF
  - we need to remove
    - all transitive dependencies i.e,. ***StudID*** → ***Dormitary***
    - The non-primary key attributes, dependent on each other i.e., ***Year*** → ***Dormitary***
  - will be moved to another table and linked with the main table using Candidate Key- Foreign Key relationship.

Third Normal Form (3NF)

STUDENT

Student_ID	StudFirstName	StudLasstName	Dept.	Year
ETS0360/13	Abel	Bekele	CS	1
ETS0365/12	Yonas	Tamru	SW	2
ETS0371/11	Lemlem	Kassa	CS	3
ETS0374/13	Fitsum	Bekele	SW	1
ETS0375/11	Aster	Alemu	SW	3

DORM

Stud_year	Dormitory
1	201
2	202
3	203

- *Generally, even though there are other four additional levels of Normalization, a table is said to be normalized if it reaches 3NF.*
- *A database with all tables in the 3NF is said to be Normalized Database.*
- *Mnemonic for remembering the rationale for normalization up to 3NF could be the following:*

1. No Repeating or Redunduncy: *no repeting fields in the table.*
2. The Fields Depend Upon the Key: *the table should solely depend on the key.*
3. The Whole Key: *no partial keybdependency.*
4. And Nothing But the Key: *no inter data dependency.*
5. So Help Me Codd: *since Codd came up with these rules.*

Other Examples on normalization

- Consider this Example:

Sales Records:

Cust Name	Item	Shipping Address	Newsletter	Supplier	Supplier Phone	Price
Alan Smith	Xbox One	35 Palm St, Miami	Xbox News	Microsoft	(800) BUY-XBOX	250
Roger Banks	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300
Evan Wilson	Xbox One, PS Vita	28 Rock Av, Denver	Xbox News, PlayStation News	Wholesale	Toll Free	450
Alan Smith	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300

Other Examples on normalization

1st Normal Form

- Each cell to be Single valued
- Entries in a column are same type
- Rows uniquely identified - Add Unique ID, or Add more colums to make unique
  (Note: The order of the rows and the order of the columns are irrelevant)

Primary Key

Cust ID	Cust Name	Item	Shipping Address	Newsletter	Supplier	Supplier Phone	Price
at_smith	Alan Smith	Xbox One	35 Palm St, Miami	Xbox News	Microsoft	(800) BUY-XBOX	250
roger25	Roger Banks	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300
wilson44	Evan Wilson	Xbox One	28 Rock Av, Denver	Xbox News	Microsoft	(800) BUY-XBOX	250
wilson44	Evan Wilson	PS Vita	28 Rock Av, Denver	PlayStation News	Sony	(800) BUY-SONY	200
am_smith	Alan Smith	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300

Other Examples on normalization

2nd Normal Form - All attributes (Non-Key Columns) dependent on the key

Primary Key

Cust ID	Cust Name	Shipping Address	Newsletter
at_smith	Alan Smith	35 Palm St, Miami	Xbox News
roger25	Roger Banks	47 Campus Rd, Boston	PlayStation News
wilson44	Evan Wilson	28 Rock Av, Denver	Xbox News
wilson44	Evan Wilson	28 Rock Av, Denver	PlayStation News
am_smith	Alan Smith	47 Campus Rd, Boston	PlayStation News

Primary Key

Item	Supplier	Supplier Phone	Price
Xbox One	Microsoft	(800) BUY-XBOX	250
PlayStation 4	Sony	(800) BUY-SONY	300
PS Vita	Sony	(800) BUY-SONY	200

Primary Key Primary Key

Cust ID	Item
at_smith	Xbox One
roger25	PlayStation 4
wilson44	Xbox One
wilson44	PS Vita
am_smith	PlayStation 4

Other Examples on normalization

3rd Normal Form - All Fields (columns) can be determined Only by the Key in the table and and no other column

Primary Key

Supplier	Supplier Phone
Microsoft	(800) BUY-XBOX
Sony	(800) BUY-SONY

Other Examples on normalization

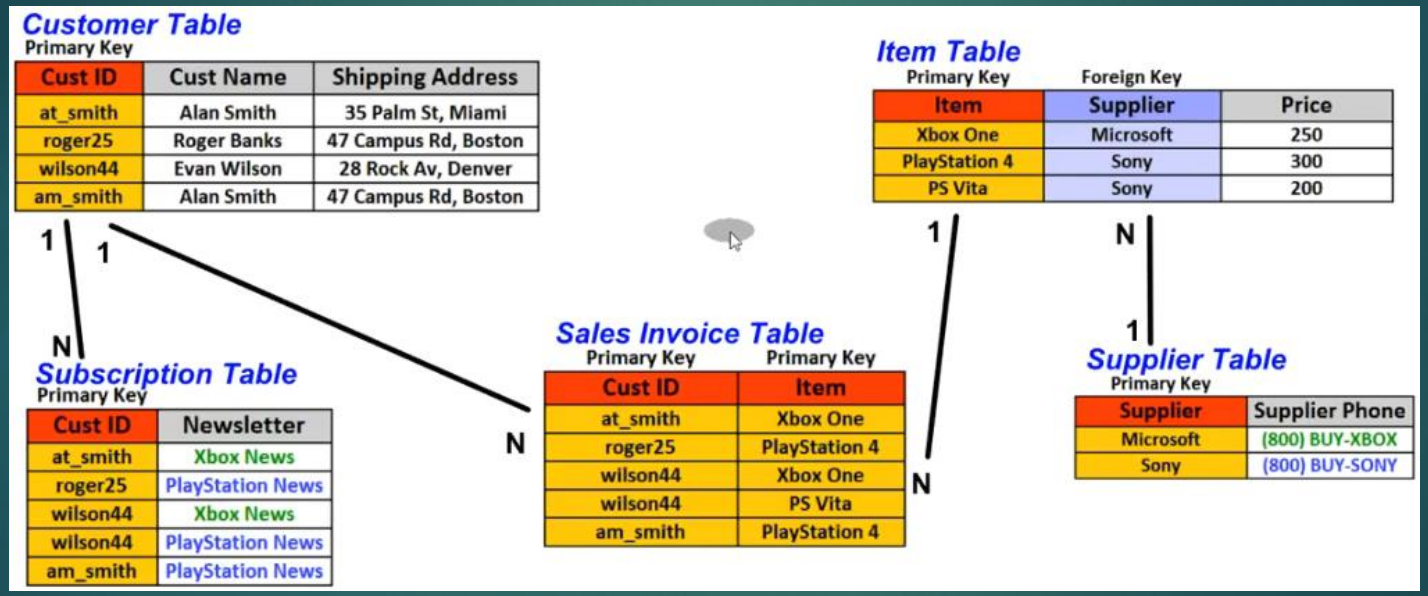
4th Normal Form - No multi-valued dependencies

Primary Key

Cust ID	Newsletter
at_smith	Xbox News
roger25	PlayStation News
wilson44	Xbox News
wilson44	PlayStation News
am_smith	PlayStation News

Other Examples on normalization

- Generally we can have this



Assignment

Relational algebra concepts