Normalization

Dongsheng Lu

4th Sem 2016

Chapter Objectives

- The purpose of normailization
- Data redundancy and Update Anomalies
- Functional Dependencies
- The Process of Normalization
- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)

Chapter Objectives (2)

- General Definition of Second and Third Normal Form
- Boyce-Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- Fifth Normal Form (5NF)

The Purpose of Normalization

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

The process of normalization is a formal method that identifies relations based on their primary or candidate keys and the functional dependencies among their attributes.

Update Anomalies

Relations that have redundant data may have problems called **update anomalies**, which are classified as ,

- Insertion anomalies
- Deletion anomalies
- Modification anomalies

Example of Update Anomalies

To insert a new staff with branchNo B007 into the StaffBranch relation;
To delete a tuple that represents the last member of staff located at a branch B007;
To change the address of branch B003.

StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St,Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St,Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St,Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Figure 1 StraffBranch relation

Example of Update Anomalies (2)

Staff

staffNo	sName	position	salary	branceNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

Branch

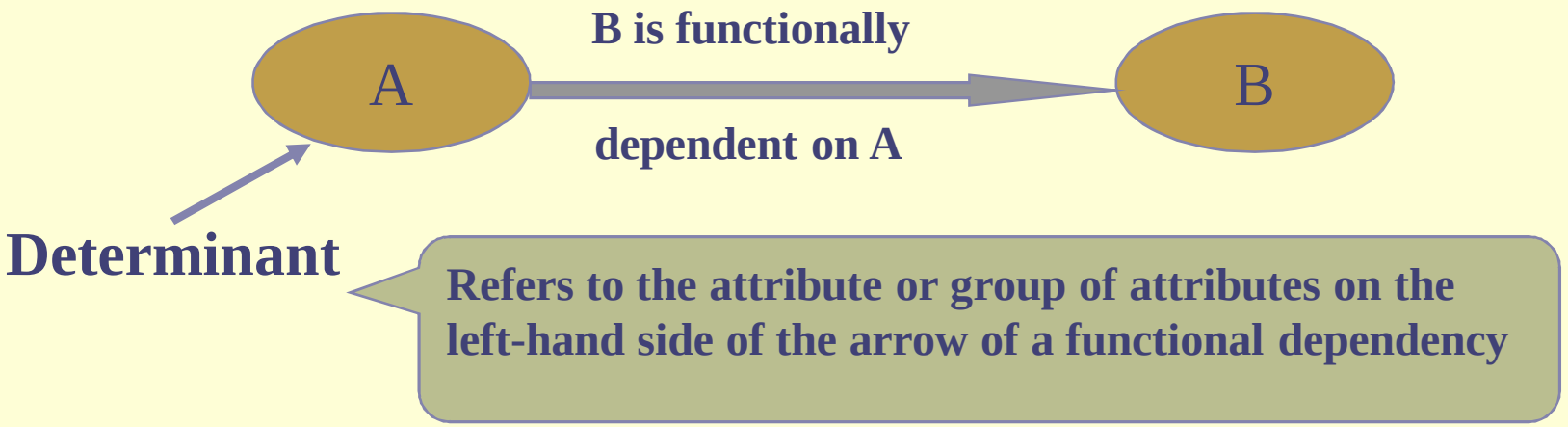
branceNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St,Glasgow

Figure 2 Straff and Branch relations

Functional Dependencies

Functional dependency describes the relationship between attributes in a relation.

For example, if A and B are attributes of relation R, and B is functionally dependent on A (denoted A $\rightarrow$ B), if each value of A is associated with exactly one value of B. (A and B may each consist of one or more attributes.)



Functional Dependencies (2)

Trival functional dependency means that the right-hand side is a subset (not necessarily a proper subset) of the left-hand side.

For example: (See Figure 1)

staffNo, sName → sName

staffNo, sName → staffNo

They do not provide any additional information about possible integrity constraints on the values held by these attributes.

We are normally more interested in **nontrivial dependencies** because they represent integrity constraints for the relation.

Functional Dependencies (3)

Main characteristics of functional dependencies in normalization

- Have a one-to-one relationship between attribute(s) on the left- and right- hand side of a dependency;
- hold for all time;
- are nontrivial.

Identifying the primary key

Functional dependency is a property of the meaning or semantics of the attributes in a relation. When a functional dependency is present, the dependency is specified as a **constraint** between the attributes.

An important integrity constraint to consider first is **the identification of candidate keys, one of which is selected to be the primary key** for the relation using functional dependency.

Inference Rules

A set of all functional dependencies that are implied by a given set of functional dependencies X is called closure of X, written X^+. A set of inference rule is needed to compute X^+ from X.

Armstrong's axioms

1. **Relfexivity:** If B is a subset of A, them A → B
2. **Augmentation:** If A → B, then A, C → B,C
3. **Transitivity:** If A → B and B → C, then A→ C
4. **Self-determination:** A → A
5. **Decomposition:** If A → B,C then A → B and A→ C
6. **Union:** If A → B and A → C, then A→ B,C
7. **Composition:** If A → B and C → D, then A,C→ B,D

Minial Sets of Functional Dependencies

A set of functional dependencies X is **minimal** if it satisfies the following condition:

- Every dependency in X has a single attribute on its right-hand side

- We cannot replace any dependency A → B in X with dependency C →B, where C is a proper subset of A, and still have a set of dependencies that is equivalent to X.

- We cannot remove any dependency from X and still have a set of dependencies that is equivalent to X.

Example of A Minial Sets of Functional Dependencies

A set of functional dependencies for the StaffBranch relation satisfies the three conditions for producing a minimal set.

staffNo → sName
staffNo → position
staffNo → salary
staffNo → branchNo
staffNo → bAddress
branchNo → bAddress
branchNo, position → salary
bAddress, position → salary

The Process of Normalization

- Normalization is often executed as a series of steps. Each step corresponds to a specific normal form that has known properties.

- As normalization proceeds, the relations become progressively more restricted in format, and also less vulnerable to update anomalies.

- For the relational data model, it is important to recognize that it is only first normal form (1NF) that is critical in creating relations. All the subsequent normal forms are optional.

First Normal Form (1NF)

Repeating group = (propertyNo, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

Unnorm

A table that contains one or more [illegible] groups.

ClientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John kay	PG4	6 lawrence St,Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 lawrence St,Glasgow	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	370	CO93	Tony Shaw
		PG16	5 Novar Dr, Glasgow	1-Nov-02	1-Aug-03	450	CO93	Tony Shaw

Figure 3 ClientRental unnormalized table

Definition of 1NF

First Normal Form is a relation in which the intersection of each row and column contains one and only one value.

There are two approaches to removing repeating groups from unnormalized tables:

1. Removes the repeating groups by entering appropriate data in the empty columns of rows containing the repeating data.

2. Removes the repeating group by placing the repeating data, along with a copy of the original key attribute(s), in a separate relation. A primary key is identified for the new relation.

1NF ClientRental relation with the first approach

The ClientRental relation is defined as follows,

ClientRental (clientNo, propertyNo, cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

With the first approach, we remove the repeating group (property rented details) by entering the appropriate client data into each row.

ClientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 lawrence St,Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 lawrence St,Glasgow	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	370	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-02	1-Aug-03	450	CO93	Tony Shaw

Figure 4 1NF ClientRental relation with the first approach

1NF ClientRental relation with the second approach

Client (clientNo, cName)

PropertyRentalOwner (clientNo, propertyNo, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

With the second approach, we remove the repeating group (property rented details) by placing the repeating data along with a copy of the original key attribute (clientNo) in a separte relation.

ClientNo	cName
CR76	John Kay
CR56	Aline Stewart

ClientNo	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	6 lawrence St,Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	5 Novar Dr, Glasgow	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	PG4	6 lawrence St,Glasgow	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	PG36	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	370	CO93	Tony Shaw
CR56	PG16	5 Novar Dr, Glasgow	1-Nov-02	1-Aug-03	450	CO93	Tony Shaw

Figure 5 1NF ClientRental relation with the second approach

Full functional dependency

Full functional dependency **indicates that if A and B are attributes of a relation, B is fully functionally dependent on A if B is functionally dependent on A, but not on any proper subset of A.**

A functional dependency A→B is **partially dependent** if there is some attributes that can be removed from A and the dependency still holds.

Second Normal Form (2NF)

Second normal form (2NF) is a relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on the primary key.

The normalization of 1NF relations to 2NF involves the removal of **partial dependencies**. If a partial dependency exists, we remove the function dependent attributes from the relation by placing them in a new relation along with a copy of their determinant.

2NF ClientRental relation

The ClientRental relation has the following functional dependencies:

fd1	clientNo, propertyNo → rentStart, rentFinish	(Primary Key)
fd2	clientNo → cName	(Partial dependency)
fd3	propertyNo → pAddress, rent, ownerNo, oName	(Partial dependency)
fd4	ownerNo → oName	(Transitive Dependency)
fd5	clientNo, rentStart → propertyNo, pAddress, rentFinish, rent, ownerNo, oName	(Candidate key)
fd6	propertyNo, rentStart → clientNo, cName, rentFinish	(Candidate key)

2NF ClientRental relationq

After removing the partial dependencies, the creation of the three new relations called Client, Rental, and PropertyOwner

Client (clientNo, cName)

Rental (clientNo, propertyNo, rentStart, rentFinish)

PropertyOwner (propertyNo, pAddress, rent, ownerNo, oName)

Client

ClientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

ClientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-00	31-Aug-01
CR76	PG16	1-Sep-02	1-Sep-02
CR56	PG4	1-Sep-99	10-Jun-00
CR56	PG36	10-Oct-00	1-Dec-01
CR56	PG16	1-Nov-02	1-Aug-03

PropertyOwner

propertyNo	pAddress	rent	ownerNo	oName
PG4	6 lawrence St,Glasgow	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	370	CO93	Tony Shaw

Figure 6 2NF ClientRental relation

Third Normal Form (3NF)

Transitive dependency

A condition where A, B, and C are attributes of a relation such that if A → B and B → C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

Third normal form (3NF)

A relation that is in first and second normal form, and in which no non-primary-key attribute is **transitively** dependent on the primary key.

The normalization of 2NF relations to 3NF involves the removal of transitive dependencies by placing the attribute(s) in a new relation along with a copy of the determinant.

3NF ClientRental relation

The functional dependencies for the Client, Rental and PropertyOwner relations are as follows:

Client

fd2	clientNo → cName	(Primary Key)

Rental

fd1	clientNo, propertyNo → rentStart, rentFinish	(Primary Key)
fd5	clientNo, rentStart → propertyNo, rentFinish	(Candidate key)
fd6	propertyNo, rentStart → clientNo, rentFinish	(Candidate key)

PropertyOwner

fd3	propertyNo → pAddress, rent, ownerNo, oName	(Primary Key)
fd4	ownerNo → oName	(Transitive Dependency)

3NF ClientRental relation

The resulting 3NF relations have the forms:

Client	(clientNo, cName)
Rental	(clientNo, propertyNo, rentStart, rentFinish)
PropertyOwner	(propertyNo, pAddress, rent, ownerNo)
Owner	(ownerNo, oName)

3NF ClientRental relation

Client

ClientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

ClientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-00	31-Aug-01
CR76	PG16	1-Sep-02	1-Sep-02
CR56	PG4	1-Sep-99	10-Jun-00
CR56	PG36	10-Oct-00	1-Dec-01
CR56	PG16	1-Nov-02	1-Aug-03

PropertyOwner

propertyNo	pAddress	rent	ownerNo
PG4	6 lawrence St,Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	370	CO93

Owner

ownerNo	oName
CO40	Tina Murphy
CO93	Tony Shaw

Figure 7 2NF ClientRental relation

Boyce-Codd Normal Form (BCNF)

Boyce-Codd normal form (BCNF)

A relation is in BCNF, if and only if, every determinant is a candidate key.

The difference between 3NF and BCNF is that for a functional dependency A → B, 3NF allows this dependency in a relation if B is a primary-key attribute and A is not a candidate key, whereas BCNF insists that for this dependency to remain in a relation, A must be a candidate key.

Fourth Normal Form (4NF)

Multi-valued dependency (MVD)
represents a dependency between attributes (for example, A, B and C) in a relation, such that for each value of A there is a set of values for B and a set of value for C. However, the set of values for B and C are independent of each other.

A multi-valued dependency can be further defined as being trivial or nontrivial. A MVD A →> B in relation R is defined as being trivial if

- B is a subset of A

or

- A U B = R

A MVD is defined as being nontrivial if neither of the above two conditions is satisfied.

Fourth Normal Form (4NF)

Fourth normal form (4NF)

A relation that is in Boyce-Codd normal form and contains no nontrivial multi-valued dependencies.

Fifth Normal Form (5NF)

Fifth normal form (5NF)

A relation that has no join dependency.

Lossless-join dependency

A property of decomposition, which ensures that no spurious tuples are generated when relations are reunited through a natural join operation.

Join dependency

Describes a type of dependency. For example, for a relation R with subsets of the attributes of R denoted as A, B, …, Z, a relation R satisfies a join dependency if, and only if, every legal value of R is equal to the join of its projections on A, B, …, Z.