Database Normalization

Saraswathy S

Objectives

- Role of Normalization.
- Understanding Data Redundancy.
- Understanding functional Dependency.
- Different Normal Forms.
- What is denormalization?

Why Normalize?

- Flexibility
  - Structure supports many ways to look at the data
- Data Integrity
  - “Modification Anomalies”
    - Deletion
    - Insertion
    - Update
- Efficiency
  - Eliminate redundant data and save space

Normalization

- The process of organizing data to minimize duplication.
- Involves dividing a database into two or more tables and defining relationships between them.
- Normalization is usually done in stages, with each stage applying more rigorous rules.
- The rules are called Normal Forms

Data Redundancy

- Redundancy is attained when the same data is stored more than once in a table, or more than one table.
- It increases the size of the database unnecessarily.
- For example, an online business may have the same customer's name appearing several times if that customer has bought several different products at different times.
- Maintaining the database is costly because any update on customer's details is done in numerous different locations.
- Worst case is, if one or more instances of the customer's name are not updated the DB become inconsistent.

Normal Forms

- A series of logical steps to take to normalize data tables.
- Edgar F. Codd originally established three normal forms: 1NF, 2NF and 3NF.
- Are sufficient for most applications.
- Most tables when reaching 3NF are also in BCNF (Boyce-Codd Normal Form).

Functional Dependency

- Let R be a relation scheme and X, Y be sets of attributes in R.
  - A functional dependency from X to Y exists if and only if:
    - For every instance of |R| of R, if two tuples in |R| agree on the values of the attributes in X, then they agree on the values of the attributes in Y
- We write X→Y and say that X determines Y
  - Example on Student (sid, name, supervisor_id, specialization):
  - {sid} → {name, supervisor_id, specialization}
    - The sid determines all attributes (i.e., the entire record)
    - If two tuples in the relation student have the same sid, then they must have the same values on all attributes.
    - In other words they must be the same tuple (since the relational modes does not allow duplicate records)

First Normal Form (1NF)

- All columns (fields) must be atomic
  - Means : no repeating items in columns
  - Ensure that each table has a primary key: minimal set of attributes which can uniquely identify a record

Unnormalized Table

EmpNo	EmpName	…	Deptno	DeptName	PrjId	PrjName	Hrs
E001	SMITH	…	10	Development	P01 P03	ERP Payroll	50 70
E002	JOHN	…	20	Testing	P04 P01	MIS ERT	35 26
..	..	..	…	..	..	..	..

First Normal Form (1NF)

- Eliminate the repeating groups into a new table.

PK

Employees

EmpNo	EmpName	…	Deptno	DeptName
E001	SMITH	…	10	Development
E002	JOHN	…	20	Testing
..	..	..	…	..

CPK

Emp_Prj

EmpNo	PrjId	PrjName	Hrs
E001	P01	ERP	50
E001	P03	Payroll	70
E002	P04	MIS	35
E002	P01	ERT	26
..	..	..	..

Second Normal Form

- Already in 1NF.
  - The database must meet all the requirements of the first normal form.
- No Partial Dependency.
  - If a table has a composite key, all attributes must be related to the whole key.
  - Can be applied only on the tables which has CPK.

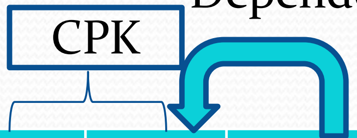
CPK

Emp_Prj

EmpNo	PrjId	PrjName	Hrs
E001	P01	ERP	50
E001	P03	Payroll	70
E002	P04	MIS	35
E002	P01	ERT	26
..	..	..	..

2 NF

- Identify and eliminate the non – key attributes that are partially dependent.
- The partial dependent key need to be made as PK of new table



Partially Dependent

CPK

EmpNo	PrjId	PrjName	Hrs
E001	P01	ERP	50
E001	P03	Payroll	70
E002	P04	MIS	35
E002	P01	ERT	26
..	..	..	..

Emp_Prj

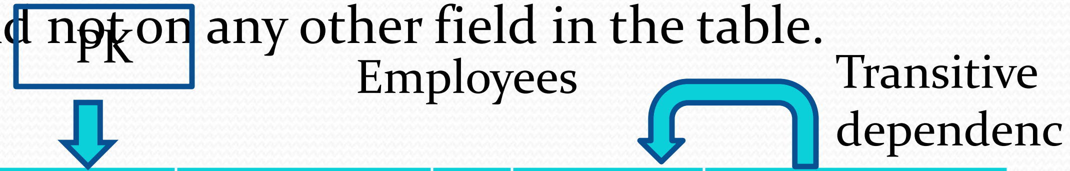
EmpNo	PrjId	Hrs
E001	P01	50
E001	P03	70
E002	P04	35
E002	P01	26
..	..	..

Projects

PrjId	PrjName
P01	ERP
P03	Payroll
P04	MIS
..	..

Third Normal Form

- Already in 2NF.
  - The database must meet all the requirements of the second normal form.
- No Transitive Dependency.
  - The data stored in a table be dependent only on the primary key, and not on any other field in the table.



Employees

EmpNo	EmpName	...	Deptno	DeptName
E001	SMITH	...	10	Development
E002	JOHN	...	20	Testing
..	..	..	...	..

3NF

- Identify and eliminate the fields which is dependent on Non-Key field.

PK

Employees

EmpNo	**EmpName**	**…**	**Deptno**
E001	SMITH	…	10
E002	JOHN	…	20
..	..	..	…

PK

Departments

Deptno	**DeptName**
10	Development
20	Testing
…	..

BCNF

- It should already be in 3NF.
- In most practical cases when a relation is in 3NF it is also in BCNF.
- Requires that the determinant of every functional dependency in a relation be a key .
- Can be applied on following scenario.
  - A table has more than 1 candidate keys.
  - Candidates are composite.
  - There is a overlapping of keys

BCNF

Candidate 1

Candidate 2

PrjId	EmpNo	PrjName	Hrs
P01	E001	ERP	50
P03	E001	Payroll	70
P04	E002	MIS	35
P01	E002	ERT	26
..	..	..	..

Emp_Prj

EmpNo	PrjId	Hrs
E001	P01	50
E001	P03	70
E002	P04	35
E002	P01	26
..	..	..

Projects

PrjId	PrjName
P01	ERP
P03	Payroll
P04	MIS
..	..

When Not to Normalize

- Want to keep tables simple so user can make their own queries
  - Avoid processing multiple tables
- Archiving Records
  - If No need to perform complex queries or "resurrect"
  - Flatten and store in one or more tables
- Testing shows Normalization has poorer performance
  - Can also try temp tables produced from Make Table queries

De-Normalization

- Many critical queries and reports exist which rely upon data from more than one table. Often times these requests need to be processed in an on-line environment.
- Repeating groups exist which need to be processed in a group instead of individually.
- When you denormalize a database, you can encounter problems with data consistency

Summary

- Normalization helps in better maintenance of the DB.
- Reduce the data redundancy.
- Normal forms are used to normalize the DB with no data loss.
- Intentional introduction of redundancy in order to improve the performance is called denormalization.