

Normalisation

Normalisation

- Main objective in developing a logical data model for relational database systems is to create an accurate and efficient representation of the data, its relationships, and constraints
- To achieve this objective, we must identify a suitable set of relations (tables)

Normalisation

- A technique for producing a set of relations with desirable properties, given the data requirements of an enterprise
- Developed by E.F. Codd (1972)
- Often performed as a series of tests on a relation to determine whether it satisfies or violates the requirements of a given *normal form (nf)*

Data Redundancy

- Major aim of relational database design is to group attributes into relations to minimize data redundancy and reduce file storage space required by base relations
- Problems associated with data redundancy are illustrated by comparing the StaffBranch relation with the Staff and Branch relations

Data Redundancy

STAFFBRANCH					
Staff_No	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

- Clearly, Branch information (BranchNo and Baddress) is repeated
- A 'better' structure is possible with two tables

Data Redundancy

STAFF					BRANCH	
Staff_No	SName	Position	Salary	BranchNo	BranchNo	Baddress
SL21	John White	Manager	30000	B005	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	B007	16 Argyll St, Aberdeen
SA9	Mary Howe	Assistant	9000	B007		
SG5	Susan Brand	Manager	24000	B003		
SL41	Julie Lee	Assistant	9000	B005		

- Reduces redundancy, decreases storage requirements
- Also other advantages - reduces the chance of update anomalies

Update Anomalies

- Relations that contain redundant information may potentially suffer from update anomalies.
- Types of update anomalies include
 - Deletion
 - Modification
 - Insertion

Update Anomalies

STAFFBRANCH					
Staff_No	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

- What happens if staff member SA9 leaves?
- What happens if branch B003 moves?
- What happens if a new branch is created (with no staff yet)?

Appropriate Design

- Clearly some designs are 'better' than others
- 'Better' here means
 - Appropriate
 - Efficient
- Moving to a better design is often called 'decomposition'
- There are properties that should hold in decompositions

Lossless-join and Dependency Preservation Properties

- Two important properties of decomposition
 - *Lossless-join property* enables us to find any instance of the original relation from corresponding instances in the smaller relations
 - *Dependency preservation property* enables us to find any relation or constraint that exists in the original relation in the new smaller relations

Lossless-Join decomposition (LJD)

STAFFBRANCH					
Staff_No	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

- Can the original information be reformed (typically with a join)?
- Consider these decompositions - are they LJDs?

Staff(Staff_No, SName, Position, Salary)

Branch(BranchNo, Baddress)

Staff(Staff_No, Sname, Position)

Branch(BranchNo, Baddress)

Employed(Position, Salary)

Dependency Preserving Decomposition (DPD)

Staff_No	SName
SL21	John White
SG37	Ann Beech
SG14	David Ford
SA9	Mary Howe
SG5	Susan Brand
SL41	Julie Lee

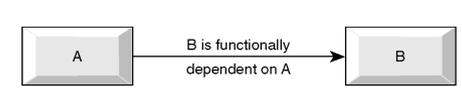
- There is a relationship between the two attributes Staff_no and Sname (called a functional dependency)
- If Staff_No and Sname are placed in two separate tables alone, the relationship is lost (and is not recoverable)
- A DPD must retain the (required) functional dependencies

Functional Dependency

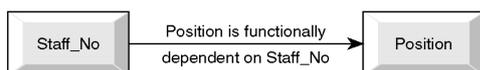
- Main concept associated with normalisation
- Functional Dependency
 - Describes the relationship between attributes in a relation
 - For example, if A and B are attributes of relation R, B is functionally dependent on A ($A \rightarrow B$), if each value of A in R is associated with exactly one value of B in R

Functional Dependency

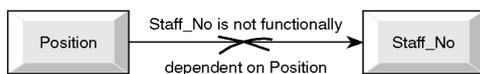
- A property of the meaning or semantics of the attributes in a relation
- The *determinant* of a functional dependency refers to the attribute or group of attributes on the left-hand side of the arrow



Example – Functional Dependency



Staff number SL21 → Manager
(a)



Manager → Staff number SL21
Manager → Staff number SG5
(b)

Exercise

Customer No	Property No	Cname	Paddress	RentStart	RentFinish	Rent	Owner_No	Oname
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-94	31-Aug-96	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-96	1-Sep-08	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-92	10-Jun-94	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-94	1-Dec-95	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Jan-96	10-Aug-96	450	CO93	Tony Shaw

- Identify the functional dependencies in the above relation
- What is the relationship between the number of attributes and number of FDs?