



Database Design

Purpose



- Optimal database design
 - Selection of appropriate relations/tables for a given set of attributes.
- Minimize update anomalies
 - Redundancy
 - Update
 - Inconsistent data
 - Additions
 - Deletions

Example

CUSTNUMB	CUSTNAME	CUSTADDR	SNUMB	SLSRNAME
123	Jones, R.	19 Oak St.	3	Adams, M.
456	Lan, J.	4 Pine St.	6	Smith, R.
461	Chu, W.	22 Main St.	12	Brown, M.
489	Obie, S.	76 High St.	6	Smith, R.
514	Wise, R	17 Birch St.	3	Adams, M.
...
999	Side, E.	87 Bay St.	12	Brown, N.

Specific Anomalies in this Relation

- Redundancy
 - <Why repeat the Slsrname for Adams each record?>
- Update
 - <suppose Slsr Mary Adams marries and changes her name?>
- Inconsistent data
 - <Notice Brown's first initial varies - M, N>
- Additions
 - <New Slsr J. Doe can't be entered until he has customer>
- Deletion
 - <Delete all customers of Adams, and we lose the name of Adams>

We Will Learn Decomposition of Relations

CUSTNUMB	CUSTNAME	CUSTADDR	SNUMB
123	Jones, R.	19 Oak St.	3
456	Lan, J.	4 Pine St.	6
461	Chu, W.	22 Main St.	12
489	Obie, S.	76 High St.	6
514	Wise, R.	17 Birch St.	3
...
999	Side, E.	87 Bay St.	12

SNUMB	SLSRNAME
3	Adams, M.
6	Smith, R.
12	Brown, M.

Notice that this Decomposition Resolved All Database Anomalies

- Redundancy
 - <None exists>
- Update
 - <Just change Mary Adams' last name (once) in salesrep relation>
- Inconsistent data
 - <Impossible - M. Brown's name appears only once!>
- Additions
 - <Add new Slsr j. Doe to salesrep relation>
- Deletions
 - <We can delete all of Adams' customers and still have Adams IN salesrep>

Decompositions



- We saw an example of decomposing a "big" relation into smaller relations with fewer attributes per relation.
- But some decompositions might create problems for us.
- There are 'good' decompositions and there are 'poor' decompositions.

Decompositions: Definitions and Properties

- A decomposition of a relation r is a set of relations $\{R_1, R_2, \dots, R_n\}$ such that each R_i is a subset of R , and the union of all r_i is R .
 - Note that the r_i may have attributes in common.
- Desirable property of decompositions - After we do a decomposition into smaller relations $\{R_1, r_2, \dots, R_n\}$, we would like, By A Join, to be able to recreate the original relation R

Normal Forms

- There are numerous "normal forms" which have been proposed to enable us to group together attributes as per our decompositions:
 - First normal form
 - Second normal form
 - Third normal form
 - Boyce-codd normal form
 - Fourth normal form
 - Domain key normal form

First Normal Forms

- A relation is in first normal form (1NF) if every attribute in every row can contain only a single value.
- A 1NF relation cannot have any row that contains a repeating grouping of attribute values.

Example of Relation Not in 1NF

ORDNUMB	ORDDTE	PARTNUMB	NUMBORD
12489	30193	AX12	11
12491	30293	BT04	1
		BZ66	1
12495	30493	CX11	2

***WE CAN CONVERT ABOVE TO 1NF BY FLATTENING ***

ORDNUMB	ORDDTE	PARTNUMB	NUMBORD
12489	30193	AX12	11
12491	30293	BT04	1
12491	30293	BZ66	1
122495	30493	CX11	2

Second Normal Form

- Definition: An attribute is a non-key attribute if It is not a part of the primary key.
- Definition: A relation is in second normal form (2NF) If it is in first normal form and no non-key attribute Is dependent on only a portion of the primary key (when the primary key is composite - Consisting of 2 or more attributes).

Example of Relation in 1NF, Not 2NF

ORDNUMB	ORDDTE	PARTNUMB	PDESC	NUMBORD	QUOPRICE
12489	90597	AX12	MOUSE	11	14.95
12491	90597	BT04	DRV5.1G	1	402.99
12491	90597	BZ66	DRV6.1G	1	311.95
12495	90797	AX12	MOUSE	4	14.95

*****FOLLOWING FD'S HOLD ON THIS RELATION*****

ORDNUMB ----> ORDDTE

PARTNUMB ---> PDESC

ORDNUMB, PARTNUMB ----> NUMBORD, QUOPRICE

*****NOT IN 2NF BECAUSE ...*****

PDESC IS DEPENDENT ON ONLY A PORTION OF PRIMARY
KEY, AND SIMILARLY FOR ORDDTE

Transform Relation to 1NF, Not 2NF

- First, take each subset of the set of attributes which make up the primary key, and begin a relation with this subset as its primary key:
 - (ORDNUMB,
 - (PARTNUMB,
 - (ORDNUMB,PARTNUMB,
- Then, place each of the other attributes with the appropriate primary key, I.E., Place each one with the minimal collection on which it depends:
 - (ORDNUMB, ORDDTE)
 - (PARTNUMB, PARTDESC)
 - (ORDNUMB, PARTNUMB, NUMBORD, QUOTEPRICE)

Third Normal Form

- A relation is in third normal form (3NF) if it is in second normal form and there is no non-key attribute which is functionally dependent upon another non-key attribute in any functional dependency.
- "Each non-key attribute must depend upon the key, the whole key, and nothing but the key“.

Example of a Relation in 2NF, Not 3NF

- Consider student(stuid, stuname, major, credits, fsjs) with the following fd's:
STUID ----> STUNAME, MAJOR, CREDITS, FSJS
CREDITS---> FSJS
- Since attribute FSJS depends on credits, student is not 3NF.
- To create 3NF here, form a new relation with the functionally dependent attribute and its determinant.
- STU2(STUID, STUNAME, MAJOR, CREDITS) R1
- STATS(CREDITS, FSJS) R2

2NF, Not 3NF or BCNF

TAGNUM	COMPID	EMPNUM	EMPNAME	LOCATION
32808	M759	611	DINH, M.	ACCOUNTING
37691	B121	124	ALVAREZ, R	SALES
57772	C007	567	FEINSTEIN, B	INFO SYSTEMS
59836	B221	124	ALVAREZ, R	HOME
77740	M759	567	FEINSTEIN, B	HOME

Some Anomalies Present in this Relation

- Update: If Betty Feinstein gets married, must change more than 1 record.
- Inconsistent data: potential problem due to redundancy.
- Additions: new employee 347 cannot be added until a pc is assigned.

Why is the PC Relation Not in 3NF or Boyce Codd Normal Form

- 1) It is in 2nf (there is no non-key attribute dependent on only a portion of the primary key, since the primary key consists of only 1 attribute).
- 2) The primary key is tagnum.
- 3) The only candidate key is tagnum.
- 4) There are 2 determinants - tagnum and empnum.
- 5) Since empnum is a determinant but not a candidate key, the relation is not in bcnf. And it's not in 3nf either.

Changing Our PC Relation to 3NF

- PC (TAGNUM, COMPID, *EMPNUM*, EMPNAME, LOCATION) is replaced by:
 - PC (TAGNUM, COMPID, EMPNUM, LOCATION)
 - and
 - EMPLOYEE(EMPNUM, EMPNAME)