

Data Redundancy & Update Anomalies

Made by

Jenish Patel
Kalpesh Patel
Kaval Patel

- ▶ A major aim of relational database design is to group attribute into relations to minimize **data redundancy** and thereby reduce the file storage space required by the implemented base relations.

Data redundancy

- ▶ **Data redundancy** is a condition created within a database or **data storage technology** in which the same piece of **data** is held in two separate places.

Data redundancy

► Let's consider following three relations :

- Staff(Staff_no, Sname, Position, Salary, Branch_no)
- Branch(Branch_no, Baddress)
- Staffbranch(Staff_no, Sname, Position, Salary, Branch_no, Baddress)

The primary keys are underlined.

► These relations contain the data given below:

1) Staff

Staff_no	Sname	Position	Salary	Branch_no
S1	John	Manager	10000	B005
S2	Ann	Assistant	5000	B003
S3	Suhas	Supervisor	7000	B007
S4	Julie	Assistant	5000	B007
S5	Mary	Assistant	5000	B005

2) Branch

Branch_no	Baddress
B005	22 deccan, Pune.
B003	10 bandra, Mumbai.
B007	32 Main st, Nasik.

3) Staffbranch

Staff_no	Sname	Position	Salary	Branch_no	Baddress
S1	John	Manager	10000	B005	22 deccan, Pune.
S2	Ann	Assistant	5000	B003	10 bandra, Mumbai.
S3	Suhas	Supervisor	7000	B007	32 Main st, Nasik.
S4	Julie	Assistant	5000	B007	32 Main st, Nasik.
S5	Mary	Assistant	5000	B005	22 deccan, Pune.

- ▶ In Staffbranch relation there is redundant data.
- ▶ Branch address is repeated for every member of staff located at that branch.
- ▶ In constant, the branch detail appear only once for each branch in the Branch relation.
- ▶ And only the Branch_no is repeated in the Staff relation.

► Relations that have redundant data have problems called **Update Anomalies**, which are classified as:

1. Insertion anomalies
2. Deletion anomalies
3. Modification anomalies

Insertion anomalies

There are two main types of insertion anomalies:

- i. To insert the detail of new staff located at the branch number B007, we must enter the correct detail of branch No. B007 in other tuple in Staffbranch relation. But table (2) doesn't suffer from this inconsistency.
- ii. To insert detail of new branch that currently has no staff into Staffbranch relation, it is necessary to enter nulls into attributes for Staff, such as Staff_no. But Staff_no violates integrity and is not allowed. The design of relations shows in table(2) avoid this problem because branch details are entered in Branch relation separately from the Staff detail.

Deletion anomalies

- ▶ If only one staff is working at a branch, and if that staff's information is deleted, then branch details are also lost, from the database.
- ▶ For example, if Staff_no S2's record is deleted, the branch information of Branch_no B003 is also lost. The design of the relations in table(2) avoid this problem, because branch tuples stored separately from Staff tuple and attribute and attribute Branch_no relates two relations.

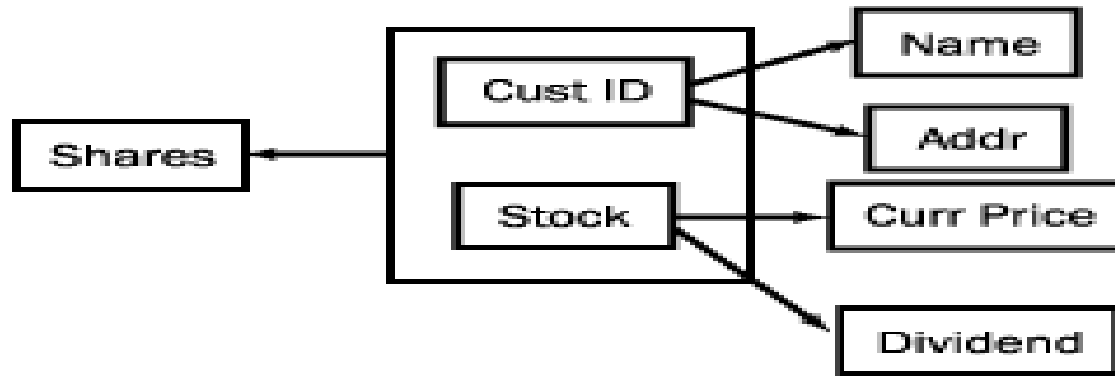
Modification anomalies

- ▶ If we want to change the value of one of the attributes of a particular branch in the Staffbranch relation.
- ▶ For example, the address for branch No. B003, we must update tuples of all staff located at the branch. If this modification is not carried out, the database relation will become inconsistent.

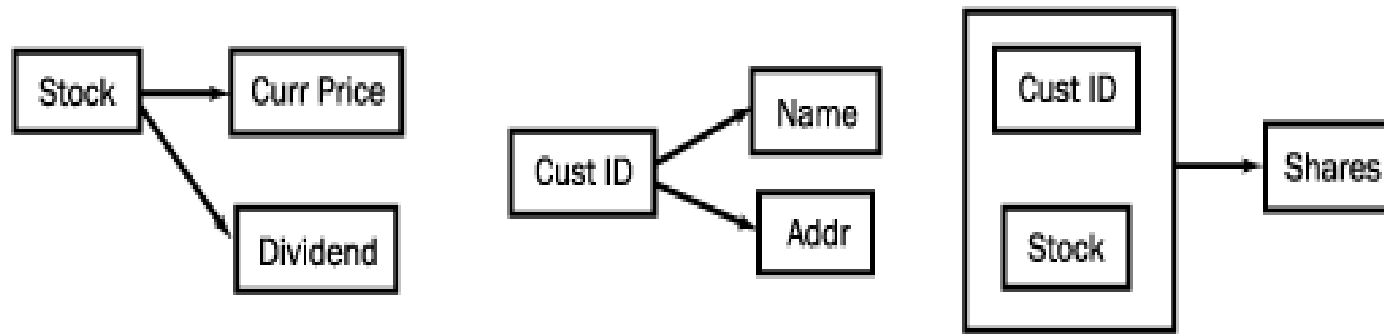
How To Avoid Anomalies??

- ▶ The use of normalization
- ▶ The goal of the normalization process is to define relations
- ▶ So that each relation is about one kind of thing.
- ▶ This seems like a reasonable condition, given the problems that it prevents

Representation of determinancy information for the stock table



Graphical representation of "well normalized" relations



Stock		
Stock	Curr Price	Dividend
IBM	100	3
C	50	2

Ownership		
Cust ID	Stock	Shares
003	IBM	16
019	C	5
102	IBM	10

Customer		
Cust ID	Name	Addr
003	Smith	LA
019	Jones	NY
102	Harris	KC

How Normalization works??

- ▶ If you know a customer id, then you know the person's name and address.
- ▶ If you know a stock identifier, then you know its current price and most recent dividend.
- ▶ Finally, for any pairing of a customer id and a stock identifier, you know how many shares that person owns of that stock

QUESTIONS ARE GUARANTEED IN LIFE
BUT ANSWERS ARE NOT...!!



ANY QUESTIONS????

THANK YOU