

Database Normalization

Christopher Simpkins

chris.simpkins@gatech.edu

Database Design

- ❏ What entities do we need to store?
- ❏ What questions (queries) do we need to be able to answer?
- ❏ Goal: minimize redundancy without losing information
Removing design flaws from a database schema
- ❏ Normalization: removing design flaws from a database schema

Data Manipulation Anomalies

emplID	name	job	deptID	dept
1	Bob	Programmer	1	Engineering
2	Alice	DBA	2	Databases
3	Kim	Programmer	1	Engineering

- ❏ **Insertion anomaly** - add new employee in engineering dept, must repeat deptID and dept
- ❏ **Deletion anomaly** - delete Alice, lose the existence of the Database dept
- ❏ **Update anomaly** - change name of Engineering to Awsomeness Dept, must change name in multiple rows

Normalization

- ❏ Functional dependency: attribute A determines attribute B, or B is dependent on A if for a given value of A, B always has a particular value
- ❏ 1NF - First Normal Form means each attribute is atomic
- ❏ 2NF - Second Normal Form means every non-key attribute is dependent on the key - the whole key for composite keys
- ❏ 3NF - Third Normal Form means that every non-key attribute is dependent only on the key, that is, there are no transitive dependencies
- ❏ “The key, the whole key, and nothing but the key.”

Not 1NF

emplID	name	job	deptID	skills
1	Bob	Programmer	1	C, Perl, Java
2	Alice	DBA	2	MySQL, PostgreSQL

❑ Not in 1NF because skills has multiple values

❑ Solution: skills table

1NF

emplID	name	job	deptID
1	Bob	Programmer	1
2	Alice	DBA	2

emplID	skill
1	C
1	Perl
1	Java
2	MySQL
2	PostgreSQL

Not 2NF

emplID	name	job	deptID	skill
1	Bob	Programmer	1	C
1	Bob	Programmer	1	Perl
2	Alice	DBA	2	MySQL

- ❑ name, job, deptID are dependent on the key (emplID, skill)
- ❑ But they're also dependent on just emplID
- ❑ So the non-key attributes aren't dependent on the whole key
- ❑ Solution: separate skills table (like before)

2NF

emplID	name	job	deptID
1	Bob	Programmer	1
2	Alice	DBA	2

emplID	skill
1	C
1	Perl
1	Java
2	MySQL
2	PostgreSQL

Not 3NF

emplID	name	job	deptID	dept
1	Bob	Programmer	1	Engineering
2	Alice	DBA	2	Databases
3	Kim	Programmer	1	Engineering

- ❑ emplID determines name, job, deptID, and dept
- ❑ But deptID also determines dept - a transitive dependency
- ❑ Solution: separate dept table

3NF

emplID	name	job	deptID
1	Bob	Programmer	1
2	Alice	DBA	2
3	Kim	Programmer	1

deptID	dept
1	Engineering
2	Databases

Conclusion

- ❑ Normalization results in more tables
- ❑ There are formal methods for normalizing using set theory, but normalization just makes sense with some practice
- ❑ More tables make queries more complex