

Introduction to Design Optimisation

CSU34041

Yvette Graham



Lecture:

- Database Design
- Design Guidelines – Central Guidelines (1-4)

Database Design

Database Design

- In previous lectures we have discussed various aspects of database design
 - The formal Relational Model
 - The Entity Relationship Model
 - Entity Relationship Diagrams
 - Mapping to a Logical Database Design
 - Relational Database Schema
- We previously also looked at Functional Dependency as a way of modelling technique. We will now revisit Functional Dependency by as a way of assessing the quality of a relation design

Database Design

- Need a formal method for analysing how relations and attributes are grouped
- A measure of appropriateness or quality, other than the intuition of the designer
 - To assess the quality of the design
- Measures
 - Design guidelines
 - Functional Dependencies
 - Normalisation

Database Guidelines

Design Guidelines

- A set of informal guidelines
 - Can be used as measures to determine the quality of a relation schema design
 - Attribute Semantics
 - Reduction of Redundancy
 - Reduction of NULLs
 - Generation of Spurious Tuples
- These measures are not always independent of one another

Attribute Semantics

- Attributes belonging to a relation have certain real-world meaning
- Semantics of a relation
 - refers to its meaning resulting from the interpretation of attribute values in a tuple
- Careful entity relationship modeling and accurate mapping to logical design help to ensure that a relational schema design has **clear meaning**

Guideline 1

Guideline 1

- Design a relation schema so that it is easy to explain its meaning
- Give relations and attributes meaningful names
- Do not combine attributes from multiple entity types and relationship types into a single relation
 - Straightforward to interpret
 - Easy to explain its meaning

Violating Guideline 1

EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn
-------	------------	-------	---------	---------	-------	----------

EMP_PROJ

<u>Ssn</u>	<u>Pnumber</u>	Hours	Ename	Pname	Plocation
------------	----------------	-------	-------	-------	-----------

- These relations violate guideline 1 by mixing attributes from distinct real-world entities
 - Employee and Department
 - Project and Employee

Reduction of Redundancy

- One goal of database schema design can be to minimise the storage space used
- Grouping attributes into relation schemas has a significant effect on storage space

EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber
-------	------------	-------	---------	---------

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn
-------	----------------	---------



EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn
-------	------------	-------	---------	---------	-------	----------

Reduction of Redundancy

- Storing merged entities in single relations leads to another problem, *update anomalies*
- Update anomalies can be classified into:
 - Insertion anomalies
 - Deletion anomalies
 - Modification anomalies

Insert Anomalies

EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn
-------	------------	-------	---------	---------	-------	----------

- To insert a new employee into EMP_DEPT it is necessary to include either:
 - all attribute values for the department that the employee works for
 - NULLs, if the employee is not yet assigned
- Consistency becomes an issue
- Inserting a new department is difficult

Deletion Anomalies

EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn
-------	------------	-------	---------	---------	-------	----------

- Deletion of Employees and Departments inextricably linked
 - If we delete the last employee currently assigned to a particular department, the information related to department is lost from the database
- This problem does not occur if using separate relations

Modification Anomalies

EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn
-------	------------	-------	---------	---------	-------	----------

- Modification makes consistency an issue
- If the manager of a department is changed
 - It is necessary to update the tuples of every employee who works for that department
 - Records can easily get out of sync
- This problem does not occur if using separate relations

Guideline 2

Guideline 2

- Design the relation schemas so that no insertion, deletion or modification anomalies are present
 - if anomalies are present, note them clearly and ensure all application programs operate correctly
- This second guideline is consistent with guideline 1

Reduction of NULLs

- If many attributes do not apply to all the tuples of a relation, you end up with many NULL values in those tuples
 - Wastes storage space
 - Can make understanding attribute meanings more difficult
 - Leads to difficulty with Joins
 - Difficulty with aggregate functions
 - COUNT and SUM

Reduction of NULLs

- A NULL value may typically have two interpretations
 - missing but inapplicable
 - Zip Code for Irish Addresses (although we do now have EirCode!)
 - missing but applicable
 - an employee's date of birth is empty
 - unknown
 - known but absent

Guideline 3

Guideline 3

- Avoid placing attributes in a relation schema whose values may frequently be NULL
 - If NULLs are unavoidable, ensure they apply in exceptional cases and not the majority of tuples
- Using space efficiently and avoiding NULL values are the main criteria for deciding upon attribute inclusion or exclusion
 - if excluded, create a separate relation for that attribute

Violating Guideline 3

- If only 15% of employees have an office, then including an Office_Number attribute in the EMPLOYEE relation would violate guideline 3
 - Instead, create an EMPLOYEE_OFFICE relation
 - This could contain the attributes
 - Ssn, Office_Number
 - A tuple is entered in the relation for all employees with an office

Generation of Spurious Tuples

- Joins across relations should only be performed on *Primary Key – Foreign Key* pairs of attributes
- If joins are performed on attributes which are not a Primary Key – Foreign Key pairing, spurious tuples are generated as a result
 - These tuples represent information which is not valid

Guideline 4

Guideline 4

- Design relation schemas so that they can be joined using equality conditions on primary key, foreign key pairs
 - This guarantees that no spurious tuples are generated by the join
- Avoid relations that contain matching attributes that are not foreign key, primary key combinations

Summary

Design Guidelines Summary

- Informal measures used to determine the quality of a relational schema design
 - Ensure that attribute semantics are easily understood
 - Reduce the redundant information in tuples
 - Reduce the number of NULL values in tuples
 - Ensure that spurious tuples are not generated by enforcing primary key, foreign key matching