Relational Database Design

pm jat @ daiict

[Relational] **Database Design as Process**

- Input:
  - Database Requirement Specifications
  - Data items and constraints

- Final Output:

  - **(Good) Relational schema**

- Measure of Goodness: <u>Minimum data redundancy</u>

Examples of <u>Bad</u> relation designs

- #1. What is the wrong in following relation in HoR schema-
  ROOM(RNO, Wing, Floor, resident1, resident2)

- Attempt Writing following queries
  - how many rooms are single seated?
  - List ID of M Tech 2015 students who are residing in hostel?

Examples of <u>Bad</u> relation designs

- #2. What is wrong if we have a single relation in XIT schema
  XIT(StudID, Name, ProgID, CPI, Pname, Intake, DID, DName)

- Rather it is better, answer all queries become easier, you do not have to perform any join at all

studid charact	name character	progid characte	cpi numeri	pname character vary	intake smallin	did chara	dname character varying(30)
101	Rahul	BCS	8.70	BTech(CS)	30	CS	Computer Engineering
102	Vikash	BEC	6.80	BTech(ECE)	40	EE	Electrical Engineering
103	Shally	BEE	7.40	BTech(EE)	40	EE	Electrical Engineering
104	Alka	BEC	7.90	BTech(ECE)	40	EE	Electrical Engineering
105	Ravi	BCS	9.30	BTech(CS)	30	CS	Computer Engineering

Examples of Bad Relation Designs

- In most cases sensing badness is easier, in some case it may not be easily visible? Consider following few cases-

Company

EMP(<u>SSN</u>, Name, DOB, DNO, Salary, DName)

WORKS_ON(<u>SSN, PNO</u>, PName, Hours)

DA-ACad

Offers(<u>AcadYear, Semester, CourseNo</u>, Credit, FacultyID)

Registers(<u>AcadYr, Sem, CourseNo</u>, Credit, <u>StudentID</u>, Grade)

TGMC:

Team(<u>TeamID</u>, PWD, MentorID, InstituteID)

Measure of "Goodness" of a relation

- Relational model defines a formal method of measuring goodness of a relation, called Normal Forms.

- Normal Forms: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF.
  - higher the Normal Form, less the redundancies are, and better a relation is.

Side Note: Goodness measure of a ERD

- Though no formal mechanism available but, at conceptual level (like in ERD), goodness can evaluated on following parameters-

  - Make sure it includes all required data items, and
  - it captures correct semantics of data items, their relationships, and cardinality and participation constraints.

Database Design approaches

- <u>Top down approach</u>: First have Conceptual model, that is most often, ER Model and map to implementation. This normally gives reasonable "good quality" relational design.

- Other approach, called as <u>bottom up approach</u>; where we begin with all data items (attributes) as a single universal relation, and based on theory of various kind of dependencies between attributes, we go through a step-wise step decomposition of the universal relation into smaller "good" relations. This can be automated too.

Database Design approaches

- Normally, in practice, mix approach is followed, that is approach 1 followed by approach 2.

- Techniques suggested in approach 2 are used for -

  - Verification purpose. Relational schema created using approach-1 are as good as ERD is. Therefore "normalization rules" can be used to measure the goodness of relations created this way.

  - Second, using the techniques, we can further refine the design, if there is some scope of improvement!

What are problems with data redundancies?

- A relation like the one shown below has redundancies.

- Redundancies are concern, primarily not because they waste storage, but because they lead to "update anomalies"

ssn	fname	salary	superssn	dno	dname	mgrssn
101	Franklin	70000	108	5	Research	101
103	Alicia	25000	102	4	Administration	102
104	Ramesh	38000	101	5	Research	101
105	Joyce	25000	101	5	Research	101
106	Ahmad	25000	102	4	Administration	102
107	John	30000	101	5	Research	101
102	Jennifer	43000	108	4	Administration	102
108	James	56000		5	Research	101

Update Anomalies

- There are three types of update anomalies-

    – Insert anomaly

    – Modification anomalies

    – Deletion anomalies

Insertion anomaly

ssn	fname	salary	superssn	dno	dname	mgrssn
101	Franklin	70000	108	5	Research	101
103	Alicia	25000	102	4	Administration	102
104	Ramesh	38000	101	5	Research	101
105	Joyce	25000	101	5	Research	101
106	Ahmad	25000	102	4	Administration	102
107	John	30000	101	5	Research	101
102	Jennifer	43000	108	4	Administration	102
108	James	56000		5	Research	101

- Try adding an employee in department no 5?

- Try adding a new department no 6

Insertion anomaly

- For above schema EMP_DEPT, to insert an employee in department 5, we need to insert all the details of department 5 and that too, exactly same as with other employees of department 5.

- If there is any discrepancy, database will become inconsistent. Whereas if we have two different relations, there is no problem like this, only the thing is to be repeated is department code

- It is difficult to insert a department without an employee.
  - We can try to insert, by keeping all employee details NUL, but since SSN is PK, this can't be NULL.

- These are examples of "Insertion Anomalies"

Deletion anomaly

- Again consider same relation.

```
ssn |   fname   | salary | superssn | dno |     dname      | mgrssn
----+-----------+--------+----------+-----+----------------+--------
101 | Franklin  |  70000 |      108 |   5 | Research       |    101
103 | Alicia    |  25000 |      102 |   4 | Administration |    102
104 | Ramesh    |  38000 |      101 |   5 | Research       |    101
105 | Joyce     |  25000 |      101 |   5 | Research       |    101
106 | Ahmad     |  25000 |      102 |   4 | Administration |    102
107 | John      |  30000 |      101 |   5 | Research       |    101
102 | Jennifer  |  43000 |      108 |   4 | Administration |    102
108 | James     |  56000 |          |   5 | Research       |    101
```

- What happens if you delete James?
- How do you delete department research?

Deletion Anomalies

- It is the example of deletion anomaly

- It is reverse of second insertion anomaly, if we delete only employee of a department then, then information of the department is also lost.

- Deletion of a department is either complex or you loose all the employees working for that department

Modification Anomalies

- Again consider same schema –

```
ssn  |   fname    |  salary | superssn | dno |      dname       | mgrssn
-----+------------+---------+----------+-----+------------------+--------
101  | Franklin   |  70000  |    108   |  5  | Research         |    101
103  | Alicia     |  25000  |    102   |  4  | Administration   |    102
104  | Ramesh     |  38000  |    101   |  5  | Research         |    101
105  | Joyce      |  25000  |    101   |  5  | Research         |    101
106  | Ahmad      |  25000  |    102   |  4  | Administration   |    102
107  | John       |  30000  |    101   |  5  | Research         |    101
102  | Jennifer   |  43000  |    108   |  4  | Administration   |    102
108  | James      |  56000  |          |  5  | Research         |    101
```

- Suppose you want to change department of John to Administration from Research. How would do it?

- How would you change name of department Administration to HR?

Modification Anomalies

- This is modification anomaly

- If you update department information of an employee but in that case department information becomes inconsistent in database.

Update Anomalies

- Three types of Update Anomalies:
  - Insertion Anomalies
  - Deletion Anomalies
  - Modification Anomalies

- Anomalies normally come from data redundancy, in most cases if we avoid it, anomalies are automatically taken care.