Relational Database Design

Instructor : Nitesh Kumar Jha

nitesjha@soa.ac.in

ITER,S'O'A(DEEMED TO BE UNIVERSITY)

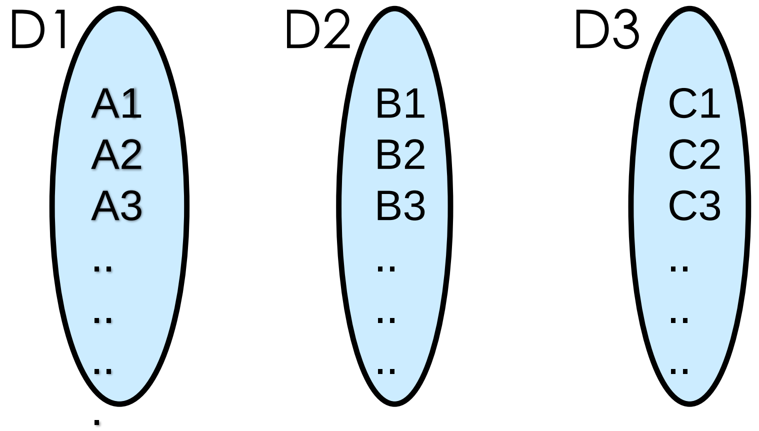
July 2018

Review

- Relational Model

Review

Domain: Set of permitted values



The Set of all possible relations: D1 ✖ D2 ✖ D3

Relation: Subset of Cartesian product of all domains.

A	B	C
A1	B3	C4
A2	B2	C3
A3	B1	C1
A4	B4	C2

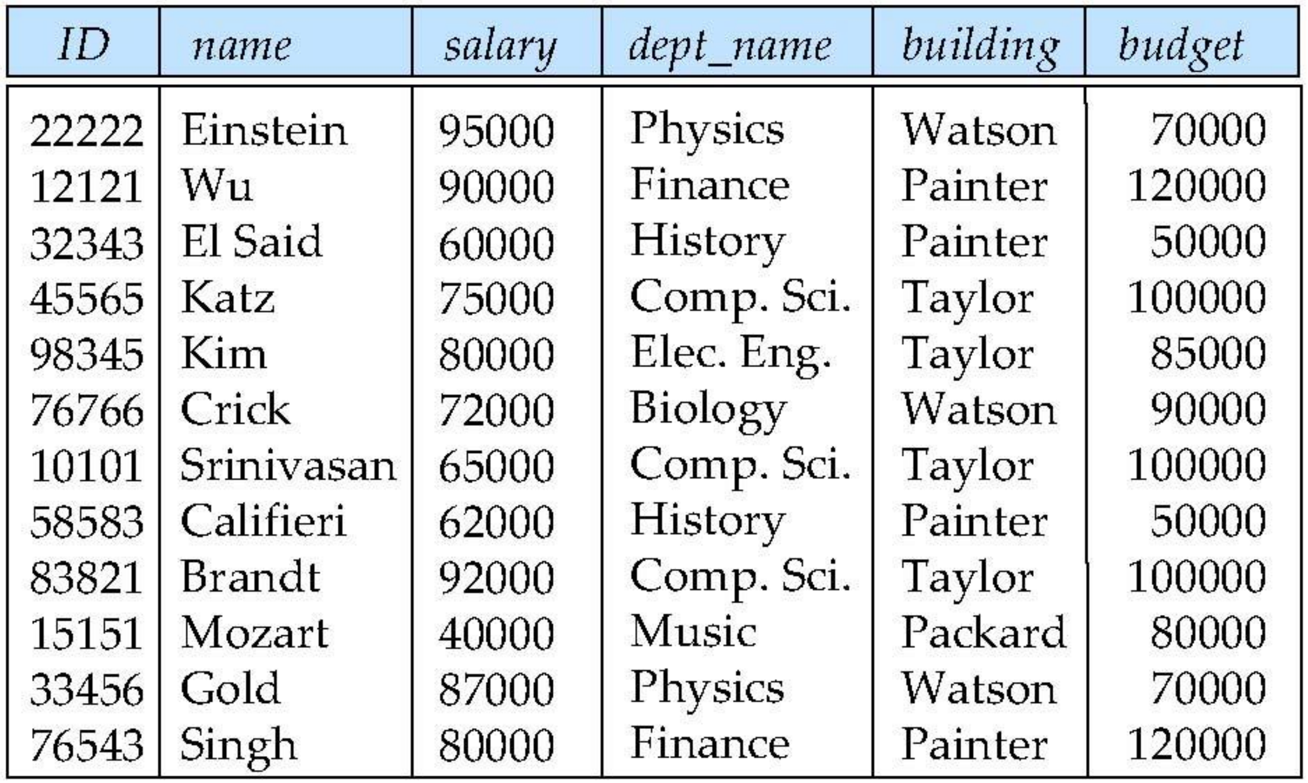
The Goal

- The goal of relational database design is to generate a set of relation schemas that allows us
  - to store information without unnecessary redundancy,
  - also allows us to retrieve information easily and efficiently.

Redundancy: The Problem

- Consider a relation schema

inst dept (ID, name, salary, dept name, building, budget)



ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Redundancy: The Problem

- Consider a relation schema

inst dept (ID, name, salary, dept name, building, budget)

- Problems
  - For each instructor of same department the building and budget information gets repeated.
  - If a new department is opened, then database is unable to keep this department information until a new instructor is appointed.
  - What is the assurance that, one department is housed in one building, and one budget?

Anomalies in Relational Database-I

- If a database not designed properly may exhibit following anomalies.
- Redundancies (repetition of information)
  - Unnecessary wastage of disk space.

studNum	Address	deptNum	deptName	Building
S21	Patna	5	CSIT	C-Block
S22	Edinburgh	5	CSIT	C-Block
S23	BBSR	4	MECH	B-Block
S24	KolKata	4	MECH	B-Block
S25	Manchester	1	PHY	D-Block

- Any change to department building information need to be updated in multiple records, that may lead to inconsistency.

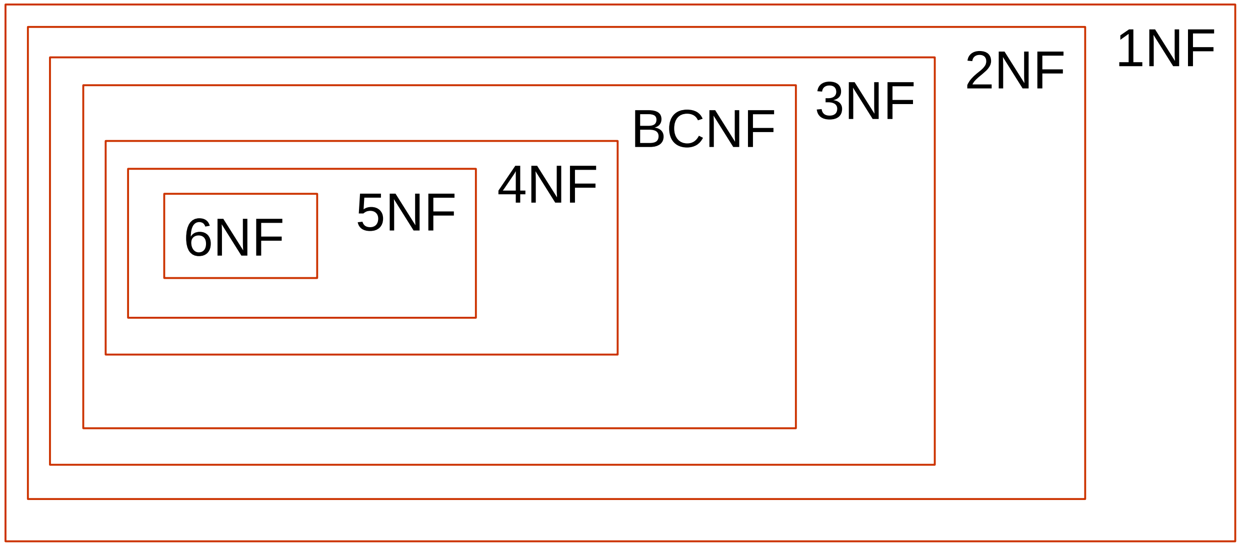
Anomalies in Relational Database

- Insertion Anomaly
  - If a new department is opened, then there is no scope to insert this information into the database unless a student gets admitted in to the department
- Deletion Anomaly
  - If the last student of a department leaves the college and hence deleted from the database, then the department information also deleted from the database forever.
- All these problems do occur due to the faulty design of the database.
- Therefore, database should be designed using normalization techniques that assures avoidance of redundancy and hence anomalies.

Solution

- The database design tries to avoid these problems using the concept of normalization
  - It is the technique of designing the relation schema in compliance to one of the several normal forms.
  - Normal forms are the well defined rules to avoid unnecessary redundancy and other anomalous conditions.

Arranged according to strictness, i.e. 6th is highest and 1st is lowest



Decomposition method

- Design Alternative: Smaller schemas
- To decompose the table: D into smaller tables D1 and D2 so that if we join the table D1 and D2 again the table should look like D.
- Finding the right decomposition is much harder for schemas with a large number of attributes and several functional dependencies. To deal with this, we shall rely on a formal methodology that we develop later in this chapter.
- Not all decompositions of schemas are helpful.
- Consider the instdept table, How can you decompose it into two schemas? Can you use Inst(ID, name, salary) and dept(name, dept_name, building, budget)? Or dept_name as common.
- How would we recognize that inst_dept requires repetition of information and should be split into the two schemas instructor and department?

Decomposition method

- A real-world database has a large number of schemas and an even larger number of attributes. The number of tuples can be in the millions or higher. Discovering repetition would be costly. There is an even more fundamental problem with this approach. It does not allow us to determine whether the lack of repetition is just a "lucky" special case or whether it is a manifestation of a general rule.
- how would we know that in our university organization, each department (identified by its department name) must reside in a single building and must have a single budget amount? We have ER model to see. However not everybody draws ER model. Therefore, we need to allow the database designer to specify rules such as "each specific value for dept name corresponds to atmost one budget" even if it is not primary key.
- In other words, we need to write a rule that says "if there were a schema (dept name, budget), then dept name is able to serve as the primary key." This rule is specified as a functional dependency

Functional Dependency (FD)

- Consider a relation schema r(R), and x and y are the subsets of R, (r: relation, R: set of attributes)
  - i.e. x, y $\subseteq$ R, then
- The instance of r(R) is said to be satisfying functional dependency $x \rightarrow y$, if for all pair of tuples t_1 and t_2
  - If $t_1[x] = t_2[x]$, then $t_1[y] = t_2[y]$
- Functional dependency $x \rightarrow y$ holds on schema *r (R) if, in* every instance of *r (R), it satisfies the functional dependency.*
- Functional dependency is a generalization of key concept of database. i.e.
- *K is a superkey of r (R) if* the functional dependency $K \rightarrow R$ *holds on r (R). (K* $\subseteq$ R*), and K uniquely determines tuples in r(R)*

Example FD

- Consider the relation schema account(accNo, balance, brID).
- There exists functional dependency like
  - accNo $\rightarrow$ balance and
  - accNo $\rightarrow$ brID, i.e
  - i.e if t1[accNo]= t2[accNo], then t1[balance] = t2[balance] etc.
- accNo uniquely determines the tuples in account relation.

Example FD

- Consider r1(A,B) with following instance of r1.

A	B
1	4
1	7
2	6
3	5

On the instance $A \rightarrow B$ does not hold but $B \rightarrow A$ holds

- Similary for r2(A,B)

What happens here??

$A \rightarrow B$???

$B \rightarrow A$???

A	B
2	5
1	3
2	5
4	7

Functional Dependency (FD)

- *K* is a super key for relation schema *R* if and only if $K \rightarrow R$
- *K* is a candidate key for *R* if and only if
  - $K \rightarrow R$, and
  - for no $\alpha \subset K$, $\alpha \rightarrow R$
- Functional dependencies allow us to express constraints that cannot be expressed using superkeys. Consider the schema:

  inst_dept (*ID, name, salary, dept_name, building, budget*).

  We expect these functional dependencies to hold:

  $$dept\_name \rightarrow building$$

  and $$ID \rightarrow building$$

  but would not expect the following to hold:

  $$dept\_name \rightarrow salary$$

Thank You