WEEK 9

GENERAL GUIDELINES FOR DATABASE DESIGN – GIVING THE REASONING BEHIND NORMALIZATION AND FUNCTIONAL DEPENDENCIES

STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
  - List 3 goals you should keep in mind when designing your relational database
  - Identity insertions, deletions and updates that cause problems with the data
  - Identify one strategy for eliminating too many null values.
  - Define the term *“lossy join” and “spurious tuples”*

GENERAL DESIGN GUIDELINES FOR DESIGNING RELATIONS

Guidelines include:

- Make sure the semantics of EACH attribute is clear in each relation (or table)
- Reducing the redundant information in tuples
- Reducing NULL values in tuples
- Disallowing the possibility of generating spurious tuples when doing joins.

extra

Goal is:

To try to characterize *good relation design* vs *bad relation design*

- **Semantics of the attributes:** Design your database so that it is easy to understand and explain the meaning and purpose for each table. Do not combine attributes from multiple entity types and relationship types.

Example – Look at this EMPLOYEE table:

Ename	SSN	Bdate	Address	Dnumber	Dname	DmgrSSN
Simpson	1243	2/2/65	2 road	S7G	Accounting	1111

QUESTION: In the above table, what 2 entity types have been combined? Should they be combined or in separate tables?

Dname & DmgrSSN.
These two tables are not necessarily needed for Simpson

ANSWER: Why would the names of the departments be stored with the employee information? We seem to be storing information about departments that don't have to do with the employee. *Make* Employee and Department separate!

11/15/23

CS319

- **Reducing the redundant values in tuples:** Design the base relations so that no insertions, deletion or modification anomalies occur in the relations.

Example:

Ename	<u>SSN</u>	Bdate	Address	Dnumber	Dname	DmgrName
Smith	23	1/1/88	4 str	S7G	Accounting	Burns
Jones	42	2/2/78	5 rd	H8Y	Payroll	Geller
Lee	11	2/2/65	2 road	S7G	Accounting	Burns

QUESTION: Give an example of an update anomaly:

Lee's department has a new name of *Accounts Payable*:
SQL is:
update EMP set Dname="Accounts Payable" where SSN="11";
QUESTION: Why is this a problem?

These two Dname are inconsist.

Ename	SSN	Bdate	Address	Dnumber	Dname	DmgrName
Smith	23	1/1/88	4 str	S7G	Accounting	Burns
Jones	42	2/2/78	5 rd	H8Y	Payroll	Geller
Lee	11	2/2/65	2 road	S7G	Accounts Payable	Burns

QUESTION: Give an example of an insertion anomaly:

QUESTION: Give an example of a deletion anomaly:

Poor old Jones is leaving our company ☹

delete from EMP where SSN="42";

QUESTION: Why is this a problem?

Ename	SSN	Bdate	Address	Dnumber	Dname	DmgrName
Smith	23	1/1/88	4 str	S7G	Accounting	Burns
Lee	11	2/2/65	2 road	S7G	Accounting	Burns
Simpson	44	2/19/79	3 str	H8Y	Payroll	Galler

we also lost pay-roll dept information!

THUS A GOAL WOULD BE:

- Design a schema that does not suffer from insertion, deletion and update anomalies.
- If there are any anomalies present, then note them so that applications can be made to take them into account.

ANOTHER GOAL:

- **Reducing the null values in tuples:** Try to avoid placing attributes in a base relation whose values may be null. If unavoidable, make sure that nulls are the exception not the majority.

Example: If only 10% of employees have a parking spot, it makes little sense to keep a parking spot attribute in the employee table, rather make a table: Emp_Parking with the Employee SSN and the Parking Spot as attributes

EMPLOYEE

Ename	SSN	Bdate	Address	P[illegible]ngSp[illegible]
Smith	23	1/1/88	4 Main str	Lev[illegible]
Jones	42	2/2/78	5 Huron rd	Null
Lee	11	2/2/65	2 Elm road	Null
Cook	33	1/1/90	4 Sun rd	Le[illegible]Sp[illegible]
Webster	26	2/2/89	77 Main str	Nu[illegible]

PARKING

ParkingSpot	EmpSSN*
Level1Spot8	23
Level2Spot6	33

ANOTHER GOAL:

- avoid schema designs that have *lossy* joins
  - Lossy joins means that they create extra (or spurious) tuples
- For this, we need to introduce the concepts of:
  - *FUNCTIONAL DEPENDENCY*
  - *NORMALIZATION*

OTHER PROBLEMS:

- Spurious Tuples

Suppose we have the table *EMPWORKSONPROJ*:

SSN	PNUMBER	HOURS	ENAME	PNAME
1	A	5	Smith	Alpha
1	B	4	Smith	Beta
2	C	10	Jones	Cappa
3	A	12	Cook	Alpha
3	B	33	Cook	Beta
4	B	23	Aziz	Beta
4	C	45	Aziz	Cappa
4	D	23	Aziz	Delta

And we split it into the following two tables:

and there's problem when we're trying to join these two tables together.

EMP_HOURS

ENAME	HOURS
Smith	5
Smith	4
Jones	10
Cook	12
Cook	33
Aziz	23
Aziz	45
Aziz	23

EMP_PROJECT

SSN	PNUMBER	ENAME	PNAME
1	A	Smith	Alpha
1	B	Smith	Beta
2	C	Jones	Cappa
3	A	Cook	Alpha
3	B	Cook	Beta
4	B	Aziz	Beta
4	C	Aziz	Cappa
4	D	Aziz	Delta

proj name lost.

Then we do a Natural Join, we will get:

SSN	PNUMBER	HOURS	ENAME	PNAME
1	A	5	Smith	Alpha
1	B	5	Smith	Beta
1	A	4	Smith	Alpha
1	B	4	Smith	Beta
2	C	10	Jones	Cappa
3	A	12	Cook	Alpha
3	B	12	Cook	Beta
3	A	33	Cook	Alpha
3	B	33	Cook	Beta
4	B	23	Aziz	Beta
…	…	…	…	…

- We know what information we expect, from this new natural join, we get too many new tuples, the join gives us **spurious tuples**. ← tuples that does not belong to it
- This is called a **lossy join**. It is lossy because we lose information (our information is no longer correct).
- We must break our big relation into relations that give us a **lossless join.**