

DB Design Concepts

DB Schema - structure of DB tables (relations) without the data

Keys for each table: attribute(s) which **uniquely determine each row** in table

Super Key → Minimal Super Keys → Candidate Keys →

Primary Key (PK) & perhaps other Unique Key(s)

Referential integrity: Foreign Key (FK) attribute(s)

FK (attribute) in table X references a PK (attribute) in table Y

(any FK value in table X refers to a unique PK row in table Y)

Design Logical DB Schema to avoid PROBLEMS (don't rely on user/programmer)

- 1) Redundancy problem - without losing data
- 2) Data anomalies problems
- 3) [Keep NULL values to a minimum (for space/time efficiency)]

EMPLOYEE_DEPARTMENT (Bad DB Schema)

empID	name	job	deptNum	deptName
7513	Nora Edwards	Programmer	128	R&D
9842	Ben Smith	DBA	42	Finance
6651	Ajay Patel	Programmer	128	R&D
9006	Candy Burnett	Sys Admin	128	R&D

Redundancy: same deptName repeated for many employees

Solution: use **Normalization** process

Anomalies:

INSERT new employee in deptNum 128 but specify Accounting (vs. R&D)

DELETE all employees in deptNum 42 → lost name of that dept

UPDATE name of deptNum 128 to Research, but miss one

Solution: use **Normalization** process

Excessive Nulls

If additional column for DeptManager, 96% NULL's, only 4% have a value

Solution: set up a separate table for DeptManager(deptNum, empID)

NORMALIZATION → put tables in ?? (1st, 2nd, 3rd) NORMAL FORM (?NF)

1NF: each attribute must be **Atomic** (a single attribute value in a cell)

EMPLOYEE (Bad DB Schema)

empID	name	job	deptNum	skill
7513	Nora Edwards	Programmer	128	C, Perl, Java
9842	Ben Smith	DBA	42	MySQL
6651	Ajay Patel	Programmer	128	VC#, Java
9006	Candy Burnett	Sys Admin	128	Windows, Linux

PROBLEM: multiple values in skill attribute for some rows

SOLUTION: see table below where PK is composite key: empID & skill

(but it's still got redundancy & anomaly problems since it's not in 2NF & 3NF)

2NF: all non-key attributes must be **FULLY Functionally Dependent** on WHOLE PK

Functional Dependencies (FD) within a table

attribute A → attribute B (A's value uniquely determines B's)

e.g., empID → name, job, deptNum, skill

EMPLOYEE (Bad DB Schema)

empID	name	job	deptNum	skill
7513	Nora Edwards	Programmer	128	C
7513	Nora Edwards	Programmer	128	Perl
7513	Nora Edwards	Programmer	128	Java
9842	Ben Smith	DBA	42	MySQL
6651	Ajay Patel	Programmer	128	VC#
6651	Ajay Patel	Programmer	128	Java
9006	Candy Burnett	Sys Admin	128	Windows
9006	Candy Burnett	Sys Admin	128	Linux

PROBLEM: empID → name, job, deptID [but empID is only part of PK, a Partial Dependency]

SOLUTION: DECOMPOSE into 2 tables :

EMPLOYEE (empID, name, job, deptNum)

EMPLOYEE_SKILL (empID, skill)

(FK: empID references EMPLOYEE table's PK: empID)

3NF: **NO Transitive Dependencies** allowed

(see table on left)

EMPLOYEE_DEPARTMENT (empID, name, job, deptNum, deptName)

PROBLEM: FD: empID → name, job, deptNum

FD: deptNum → deptName

SOLUTION: DECOMPOSE into 2 tables:

EMPLOYEE (empID, name, job, deptNum)

DEPARTMENT (deptNum, deptName)