Relational Normalization Theory
Designing Databases Using Normalization

On No!!! There’s another way to design a good set of tables besides Conceptual and Logical Modeling.

FDs, Keys, 1NF, 2NF, 3NF, BCNF, MVDs, F+, A+, FD Preservation, Lossless Join …

See also http://en.wikipedia.org/wiki/Database_normalization
A Badly Designed Table*

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
<th>DNAME</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>824</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>20-FEB-81</td>
<td>1600</td>
<td>300</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>7698</td>
<td>22-FEB-81</td>
<td>1250</td>
<td>500</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7566</td>
<td>JONES</td>
<td>MANAGER</td>
<td>7839</td>
<td>02-APR-81</td>
<td>2975</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7654</td>
<td>MARTIN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>28-SEP-81</td>
<td>1250</td>
<td>1400</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7698</td>
<td>BLAKE</td>
<td>MANAGER</td>
<td>7839</td>
<td>01-MAY-81</td>
<td>2850</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7782</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>7839</td>
<td>09-JUN-81</td>
<td>2450</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7788</td>
<td>SCOTT</td>
<td>ANALYST</td>
<td>7566</td>
<td>09-DEC-82</td>
<td>3000</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7839</td>
<td>KING</td>
<td>none</td>
<td>17-NOV-81</td>
<td>5000</td>
<td></td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN</td>
<td>7698</td>
<td>08-SEP-81</td>
<td>1500</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK</td>
<td>7788</td>
<td>12-JAN-83</td>
<td>1100</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7900</td>
<td>JAMES</td>
<td>CLERK</td>
<td>7698</td>
<td>03-DEC-81</td>
<td>950</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST</td>
<td>7566</td>
<td>03-DEC-81</td>
<td>3000</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7934</td>
<td>MILLER</td>
<td>CLERK</td>
<td>7782</td>
<td>23-JAN-82</td>
<td>1300</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
</tbody>
</table>

What’s wrong with this table?

• **Insertion Anomalies**
  
  Can’t insert information about Department 40 unless it has an employee. So where do you store information about Department 40?

• **Deletion Anomalies**
  
  If you delete EMPNOs 7782, 7839, and 7934, you lose all information about Department 10.

• **Update Anomalies**
  
  If you change CLARK’s DNAME to ‘SALES’ and leave his DEPTNO=10, you need to make sure the DNAME for DEPT 10 is changed in every tuple.

---

*This table was create with the following SQL: create table emp_dept as (select e.*, dname, loc from emp e, dept d where e.deptno = d.deptno)*


What does “No Joins” Mean in MongoDB?

http://www.slideshare.net/amitthakkar01/get-expertise-with-mongo-db

Is This a Good Thing?
A Simple Formal System

If

\[
\begin{align*}
    a &\rightarrow a\ b\ c\ d\ e\ f \\
    d &\rightarrow e\ f
\end{align*}
\]

Then

\[
\begin{align*}
    a &\rightarrow a\ b\ c\ d \\
    d &\rightarrow e\ f
\end{align*}
\]

If

\[
\begin{align*}
    a\ b &\rightarrow a\ b\ c\ d\ e\ f \\
    a &\rightarrow c\ d \\
    b &\rightarrow e\ f
\end{align*}
\]

Then

\[
\begin{align*}
    a\ b &\rightarrow a\ b \\
    a &\rightarrow c\ d \\
    b &\rightarrow e\ f
\end{align*}
\]
## A Badly Designed Table*

*This table was create with the following SQL: create table emp_dept as (select e.*, dname, loc from emp e, dept d where e.deptno = d.deptno)*

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
<th>DNAME</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>824</td>
<td></td>
<td></td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>20-FEB-81</td>
<td>1600</td>
<td>300</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>7698</td>
<td>22-FEB-81</td>
<td>1250</td>
<td>500</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7566</td>
<td>JONES</td>
<td>MANAGER</td>
<td>7839</td>
<td>02-APR-81</td>
<td>2975</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7654</td>
<td>MARTIN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>28-SEP-81</td>
<td>1250</td>
<td>1400</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7698</td>
<td>BLAKE</td>
<td>MANAGER</td>
<td>7839</td>
<td>01-MAY-81</td>
<td>2850</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7782</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>7839</td>
<td>09-JUN-81</td>
<td>2450</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7788</td>
<td>SCOTT</td>
<td>ANALYST</td>
<td>7566</td>
<td>09-DEC-82</td>
<td>3000</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7839</td>
<td>KING</td>
<td>none</td>
<td>7698</td>
<td>01-MAY-81</td>
<td>5000</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN</td>
<td>7698</td>
<td>08-SEP-81</td>
<td>1500</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK</td>
<td>7788</td>
<td>12-JAN-83</td>
<td>1100</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7900</td>
<td>JAMES</td>
<td>CLERK</td>
<td>7698</td>
<td>03-DEC-81</td>
<td>950</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST</td>
<td>7566</td>
<td>03-DEC-81</td>
<td>3000</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7934</td>
<td>MILLER</td>
<td>CLERK</td>
<td>7782</td>
<td>23-JAN-82</td>
<td>1300</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
</tbody>
</table>

In this table **EMPNO --> ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO, DNAME, LOC**

and **DEPTNO --> DNAME, LOC**

--> stands for “Funcionally Determines” which we will discuss later.
Apply the Simple Formal System

If

\[ R = \text{emp} \_\text{dept \ Relation (Table)} \]

\[
\begin{align*}
\text{EMPNO} & \rightarrow \text{ENAME \ JOB \ MGR \ HIREDATE \ SAL \ COMM \ DEPTNO \ DNAME \ LOC} \\
\text{DEPTNO} & \rightarrow \text{DNAME \ LOC}
\end{align*}
\]

Then

\[ R_1 = \text{emp \ Relation (Table)} \]

\[
\begin{align*}
\text{EMPNO} & \rightarrow \text{ENAME \ JOB \ MGR \ HIREDATE \ SAL \ COMM \ DEPTNO}
\end{align*}
\]

\[ R_2 = \text{dept \ Relation (Table)} \]

\[
\begin{align*}
\text{DEPTNO} & \rightarrow \text{DNAME \ LOC}
\end{align*}
\]

So, using the Simple Formal System, we decomposed a bad table \( R \) (which is said to not be in 3NF) into a set of tables \( R_1 \) and \( R_2 \) that are in 3NF. (The formal definition of 3NF will be given later.)
A Side Discussion of Formal Systems
(This and the next slides with green background are optional)

At the turn of the twentieth Century, mathematicians asked “Why wouldn’t it be true” that all theorems in arithmetic can be proven by a formal system. (This question was implicit in the actual problem put forth by David Hilbert in 1900 – “Prove that the axioms of arithmetic are consistent. “

This lead to 30 years of work and a three volume publication - “Principia Mathematica”
In 1930, Kurt Gödel answered the “Why wouldn’t it be true” question with his “First Incompleteness Theorem”

As part of his Incompleteness Theorem, Gödel translated the paradoxical statement:

"This statement cannot be proved"

into the pure mathematical statement:

\[ \sim(3r:3s: (P(r,s) \lor s=g(sub (f_2(y)))))) \]

and used this to show there are some mathematical statements which are true but which nevertheless cannot be proved.

First Incompleteness Theorem: There are true statements in mathematics that no formal system can prove true and there are false statements in mathematics that any formal system will prove true.

Second Incompleteness Theorem: No formal system can prove its own consistency.

In my opinion, one of the two* greatest discoveries of humankind.

*The second is John Stewart Bell’s inequality Theorem, see the book “The Age of Entanglement”.

See also Gödel's papers on the class website
Good Books to Have for a Happy Life 😊

From Frege to Gödel:

- Lambda-Calculus and Combinators: An Introduction
- A Profile of Mathematical Logic
- Gödel’s Proof
- Gödel, Escher, Bach: An Eternal Golden Braid

My Favorite all books I’ve ever read in over 65 years.
Functional Dependencies

Informal definition of a Functional Dependency

• Require that the value for a certain set of attributes uniquely determines the value for another set of attributes.

• A functional dependency is a generalization of the notion of a key.

Formal definition of a Functional Dependency

Let $R$ be a relation and

$\alpha \subseteq R$ and $\beta \subseteq R$

The functional dependency

$\alpha \rightarrow \beta$

holds on $R$ if and only if for any tuples $t_1$ and $t_2$ that agree on the attributes $\alpha$, they also agree on the attributes $\beta$.

That is, $t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$
Keys

• **Superkey** - $K$ is a superkey for relation $R$ if and only if $K \rightarrow R$

• **Candidate Key** - $K$ is a candidate key for $R$ if and only if $K \rightarrow R$ (i.e., $K$ is a super key), and for no $\alpha \subset K$, $\alpha \rightarrow R$
  
  (i.e., $K$ is minimal with respect to the number of attributes of which it is composed)

• **Prime attributes** - attributes that belong to any candidate key

• **Primary Key** – Pick one from the candidate keys
3NF - Relation R is in 3NF if:

- R is in 2NF and
- No nonprime attribute functionally determines any other nonprime attribute
## A Badly Designed Table*

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
<th>DNAME</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>824</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>824</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>20-FEB-81</td>
<td>1600</td>
<td>300</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>20-FEB-81</td>
<td>1600</td>
<td>300</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>7698</td>
<td>22-FEB-81</td>
<td>1250</td>
<td>500</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>7698</td>
<td>22-FEB-81</td>
<td>1250</td>
<td>500</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7566</td>
<td>JONES</td>
<td>MANAGER</td>
<td>7839</td>
<td>02-APR-81</td>
<td>2975</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7566</td>
<td>JONES</td>
<td>MANAGER</td>
<td>7839</td>
<td>02-APR-81</td>
<td>2975</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7654</td>
<td>MARTIN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>28-SEP-81</td>
<td>1250</td>
<td>1400</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7654</td>
<td>MARTIN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>28-SEP-81</td>
<td>1250</td>
<td>1400</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7698</td>
<td>BLAKE</td>
<td>MANAGER</td>
<td>7839</td>
<td>01-MAY-81</td>
<td>2850</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7698</td>
<td>BLAKE</td>
<td>MANAGER</td>
<td>7839</td>
<td>01-MAY-81</td>
<td>2850</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7782</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>7839</td>
<td>09-JUN-81</td>
<td>2450</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7782</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>7839</td>
<td>09-JUN-81</td>
<td>2450</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7788</td>
<td>SCOTT</td>
<td>ANALYST</td>
<td>7566</td>
<td>09-DEC-82</td>
<td>3000</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7788</td>
<td>SCOTT</td>
<td>ANALYST</td>
<td>7566</td>
<td>09-DEC-82</td>
<td>3000</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7839</td>
<td>KING</td>
<td>none</td>
<td>none</td>
<td>17-NOV-81</td>
<td>5000</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7839</td>
<td>KING</td>
<td>none</td>
<td>none</td>
<td>17-NOV-81</td>
<td>5000</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN</td>
<td>7698</td>
<td>08-SEP-81</td>
<td>1500</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN</td>
<td>7698</td>
<td>08-SEP-81</td>
<td>1500</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK</td>
<td>7788</td>
<td>12-JAN-83</td>
<td>1100</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK</td>
<td>7788</td>
<td>12-JAN-83</td>
<td>1100</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7900</td>
<td>JAMES</td>
<td>CLERK</td>
<td>7698</td>
<td>03-DEC-81</td>
<td>950</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7900</td>
<td>JAMES</td>
<td>CLERK</td>
<td>7698</td>
<td>03-DEC-81</td>
<td>950</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST</td>
<td>7566</td>
<td>03-DEC-81</td>
<td>3000</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST</td>
<td>7566</td>
<td>03-DEC-81</td>
<td>3000</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7934</td>
<td>MILLER</td>
<td>CLERK</td>
<td>7782</td>
<td>23-JAN-82</td>
<td>1300</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7934</td>
<td>MILLER</td>
<td>CLERK</td>
<td>7782</td>
<td>23-JAN-82</td>
<td>1300</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
</tbody>
</table>

*This table was create with the following SQL:

```
insert into emp_dept (select t.*, dname, loc from (select EMPNO, ENAME, JOB, MGR,HIREDATE, SAL, COMM, case
when deptno = 10 then 20
when deptno = 20 then 30
when deptno = 30 then 10
end deptno from emp) t, dept d where t.deptno = d.deptno)
```
A Badly Designed Table

What’s wrong with this table?

- **Insertion Anomalies**
  
  Can’t insert information about Department 40 unless it has an employee. So where do you store information about Department 40?

- **Deletion Anomalies**
  
  If you delete EMPNOs in department 10, you lose all information about Department 10.

- **Update Anomalies**
  
  If change CLARK’s DNAME to ‘SALES’ where DEPTNO = 10 and leave his DEPTNO=10, you need to make sure the DNAME for DEPT 10 is changed in every tuple.
A Badly Designed Table

### In this table

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
<th>DNAME</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>824</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>824</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>20-FEB-81</td>
<td>1600</td>
<td>300</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>20-FEB-81</td>
<td>1600</td>
<td>300</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>7698</td>
<td>22-FEB-81</td>
<td>1250</td>
<td>500</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>7698</td>
<td>22-FEB-81</td>
<td>1250</td>
<td>500</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7566</td>
<td>JONES</td>
<td>MANAGER</td>
<td>7839</td>
<td>02-APR-81</td>
<td>2975</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7566</td>
<td>JONES</td>
<td>MANAGER</td>
<td>7839</td>
<td>02-APR-81</td>
<td>2975</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7654</td>
<td>MARTIN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>28-SEP-81</td>
<td>1250</td>
<td>1400</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7654</td>
<td>MARTIN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>28-SEP-81</td>
<td>1250</td>
<td>1400</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7698</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>7839</td>
<td>01-MAY-81</td>
<td>2850</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7698</td>
<td>BLAKE</td>
<td>MANAGER</td>
<td>7839</td>
<td>01-MAY-81</td>
<td>2850</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7782</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>7839</td>
<td>09-JUN-81</td>
<td>2450</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7782</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>7839</td>
<td>09-JUN-81</td>
<td>2450</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7788</td>
<td>SCOTT</td>
<td>ANALYST</td>
<td>7566</td>
<td>09-DEC-82</td>
<td>3000</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7788</td>
<td>SCOTT</td>
<td>ANALYST</td>
<td>7566</td>
<td>09-DEC-82</td>
<td>3000</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7839</td>
<td>KING</td>
<td>none</td>
<td>17-NOV-81</td>
<td>5000</td>
<td></td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7839</td>
<td>KING</td>
<td>none</td>
<td>17-NOV-81</td>
<td>5000</td>
<td></td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN</td>
<td>7698</td>
<td>08-SEP-81</td>
<td>1500</td>
<td></td>
<td>1500</td>
<td>10 ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN</td>
<td>7698</td>
<td>08-SEP-81</td>
<td>1500</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK</td>
<td>7788</td>
<td>12-JAN-83</td>
<td>1100</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK</td>
<td>7788</td>
<td>12-JAN-83</td>
<td>1100</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7900</td>
<td>JAMES</td>
<td>CLERK</td>
<td>7698</td>
<td>03-DEC-81</td>
<td>950</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7900</td>
<td>JAMES</td>
<td>CLERK</td>
<td>7698</td>
<td>03-DEC-81</td>
<td>950</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST</td>
<td>7566</td>
<td>03-DEC-81</td>
<td>3000</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST</td>
<td>7566</td>
<td>03-DEC-81</td>
<td>3000</td>
<td></td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>7934</td>
<td>MILLER</td>
<td>CLERK</td>
<td>7782</td>
<td>23-JAN-82</td>
<td>1300</td>
<td></td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>7934</td>
<td>MILLER</td>
<td>CLERK</td>
<td>7782</td>
<td>23-JAN-82</td>
<td>1300</td>
<td></td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
</tbody>
</table>
### A Simple Formal System

<table>
<thead>
<tr>
<th>If</th>
<th>If</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a -&gt; a b c d e f</strong></td>
<td><strong>a b -&gt; a b c d e f</strong></td>
</tr>
<tr>
<td><strong>d -&gt; e f</strong></td>
<td><strong>a -&gt; c d</strong></td>
</tr>
<tr>
<td></td>
<td><strong>b -&gt; e f</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Then</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a -&gt; a b c d</strong></td>
<td><strong>a b -&gt; a b</strong></td>
</tr>
<tr>
<td></td>
<td><strong>a -&gt; c d</strong></td>
</tr>
<tr>
<td></td>
<td><strong>b -&gt; e f</strong></td>
</tr>
</tbody>
</table>
Apply the Simple Formal System

If

\[ R = \text{emp\_dept Relation (Table)} \]

Then

R

\[
\begin{align*}
\text{EMPNO DEPTNO} & \rightarrow \text{ENAME JOB MGR HIREDATE SAL COMM DEPTNO DNAME LOC} \\
\text{EMPNO} & \rightarrow \text{ENAME JOB MGR HIREDATE SAL COMM} \\
\text{DEPTNO} & \rightarrow \text{DNAME LOC}
\end{align*}
\]

R1

\[
\begin{align*}
\text{EMPNO DEPTNO} & \rightarrow \text{EMPNO DEPTNO}
\end{align*}
\]

R2

\[
\begin{align*}
\text{EMPNO} & \rightarrow \text{ENAME JOB MGR HIREDATE SAL COMM}
\end{align*}
\]

R3

\[
\begin{align*}
\text{DEPTNO} & \rightarrow \text{DNAME LOC}
\end{align*}
\]

R1

R2 = emp Relation (Table) without DEPTNO FK

R3 = dept Relation (Table)

So, using the Simple Formal System, we decomposed a bad table R (which is said to not be in 2NF) into a set of tables R1, R2 and R that are in 2NF. (The formal definition of 2NF will be given later.)
2NF

2NF - Relation R is in 2NF if:

R is in 1NF and

- there is no nonprime attribute (i.e., an attribute that’s not part of any candidate key) that is dependent on any part of the candidate key

- (no nonprime attribute is functionally determined by a prime attribute)

- in other words, each nonprime attribute in relation R is fully dependent upon every candidate key

Note: If relation R has no compound keys, R is automatically in 2NF
1NF

1NF - Relation R is in 1NF if:

• All data stored at the intersection of a row (tuple) and column (attribute) must be atomic (i.e., it can’t be decomposed into multiple values). See next pages for examples.

• A table (Relation) must not contain any repeating columns (attributes). See next pages for examples.
1NF – All values must be Atomic

Badly Designed Table

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK, Another Job</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>824</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN, Another Job</td>
<td>7698</td>
<td>20-FEB-81</td>
<td>1600</td>
<td>300</td>
<td>30</td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN, Another Job</td>
<td>7698</td>
<td>22-FEB-81</td>
<td>1250</td>
<td>500</td>
<td>30</td>
</tr>
<tr>
<td>7566</td>
<td>JONES</td>
<td>MANAGER, Another Job</td>
<td>7839</td>
<td>02-APR-81</td>
<td>2975</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7654</td>
<td>MARTIN</td>
<td>SALESMAN, Another Job</td>
<td>7698</td>
<td>28-SEP-81</td>
<td>1250</td>
<td>1400</td>
<td>30</td>
</tr>
<tr>
<td>7698</td>
<td>BLAKE</td>
<td>MANAGER, Another Job</td>
<td>7839</td>
<td>01-MAY-81</td>
<td>2850</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7782</td>
<td>CLARK</td>
<td>MANAGER, Another Job</td>
<td>7839</td>
<td>09-JUN-81</td>
<td>2450</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>7788</td>
<td>SCOTT</td>
<td>ANALYST, Another Job</td>
<td>7566</td>
<td>09-DEC-82</td>
<td>3000</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7839</td>
<td>KING</td>
<td>none, Another Job</td>
<td>7698</td>
<td>17-NOV-81</td>
<td>5000</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN, Another Job</td>
<td>7698</td>
<td>08-SEP-81</td>
<td>1500</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK, Another Job</td>
<td>7788</td>
<td>12-JAN-83</td>
<td>1100</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7900</td>
<td>JAMES</td>
<td>CLERK, Another Job</td>
<td>7698</td>
<td>03-DEC-81</td>
<td>950</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST, Another Job</td>
<td>7566</td>
<td>03-DEC-81</td>
<td>3000</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7934</td>
<td>MILLER</td>
<td>CLERK, Another Job</td>
<td>7782</td>
<td>23-JAN-83</td>
<td>1300</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

The Fix

Or, in some cases, create a table with a “type” column that represents a hierarchy as in the Business Contracts “Contact” table →.
1NF – No Repeating Columns (Attributes)

Badly Designed Table

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>JOB2</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>Another Job</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>824</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>Another Job</td>
<td>7698</td>
<td>20-FEB-81</td>
<td>1600</td>
<td>300</td>
<td>30</td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>Another Job</td>
<td>7698</td>
<td>22-FEB-81</td>
<td>1250</td>
<td>500</td>
<td>30</td>
</tr>
<tr>
<td>7566</td>
<td>JONES</td>
<td>MANAGER</td>
<td>Another Job</td>
<td>7839</td>
<td>02-APR-81</td>
<td>2975</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7654</td>
<td>MARTIN</td>
<td>SALESMAN</td>
<td>Another Job</td>
<td>7698</td>
<td>28-SEP-81</td>
<td>1250</td>
<td>1400</td>
<td>30</td>
</tr>
<tr>
<td>7698</td>
<td>BLAKE</td>
<td>MANAGER</td>
<td>Another Job</td>
<td>7839</td>
<td>01-MAY-81</td>
<td>2850</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7782</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>Another Job</td>
<td>7839</td>
<td>09-JUN-81</td>
<td>2450</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>7839</td>
<td>KING</td>
<td>none</td>
<td>Another Job</td>
<td></td>
<td></td>
<td>5000</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN</td>
<td>Another Job</td>
<td>7698</td>
<td>08-SEP-81</td>
<td>1500</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK</td>
<td>Another Job</td>
<td>7788</td>
<td>12-JAN-83</td>
<td>1100</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7900</td>
<td>JAMES</td>
<td>CLERK</td>
<td>Another Job</td>
<td>7698</td>
<td>03-DEC-81</td>
<td>950</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST</td>
<td>Another Job</td>
<td>7566</td>
<td>03-DEC-81</td>
<td>3000</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7934</td>
<td>MILLER</td>
<td>CLERK</td>
<td>Another Job</td>
<td>7782</td>
<td>23-JAN-82</td>
<td>1300</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Fix

Or, in some cases, create a table with a “type” column that represents a hierarchy as in the Business Contracts “Contact” table."
Normalization Example

Given the Relation $R=ABCDEF$ that’s in 1NF and the following set $F$ of Functional Dependencies,

- $A \rightarrow BC$
- $D \rightarrow EF$
- $E \rightarrow F$

find a set of Relations that are in 3NF.

You might need the following to find the Primary Key of $R$.

Armstrong’s Axioms:

- if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ **(reflexivity)** (e.g., if $\alpha = AB$ then $AB \rightarrow A$ and $AB \rightarrow B$)
  - (these are the trivial dependencies)
- if $\alpha \rightarrow \beta$, then $\gamma \alpha \rightarrow \beta$ **(augmentation)**
  - and $\gamma \alpha \rightarrow \gamma \beta$
- if $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$ **(transitivity)**

Additional rules implied by Armstrong’s Axioms:

- if $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta \gamma$ holds **(union)**
- if $\alpha \rightarrow \beta \gamma$ holds, then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds **(decomposition)**
- if $\alpha \rightarrow \beta$ holds and $\gamma \beta \rightarrow \delta$ holds, then $\alpha \gamma \rightarrow \delta$ holds **(pseudotransitivity)**
**Application Programming Interface (API)**

For Online Transaction Processing (OLTP), which this course is mainly about.

**Ad hoc Query Language**

---

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
<th>DNAME</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>824</td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
<td></td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>20-FEB-81</td>
<td>1600</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
<td></td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>7698</td>
<td>22-FEB-81</td>
<td>1250</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
<td></td>
</tr>
<tr>
<td>7566</td>
<td>JONES</td>
<td>MANAGER</td>
<td>7689</td>
<td>02-APR-81</td>
<td>2975</td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
<td></td>
</tr>
<tr>
<td>7654</td>
<td>MARTIN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>28-SEP-81</td>
<td>1250</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
<td></td>
</tr>
<tr>
<td>7698</td>
<td>BLAKE</td>
<td>MANAGER</td>
<td>7839</td>
<td>01-MAY-81</td>
<td>2850</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
<td></td>
</tr>
<tr>
<td>7782</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>7839</td>
<td>09-JUN-81</td>
<td>2450</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
<td></td>
</tr>
<tr>
<td>7788</td>
<td>SCOTT</td>
<td>ANALYST</td>
<td>7566</td>
<td>09-DEC-82</td>
<td>3000</td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
<td></td>
</tr>
<tr>
<td>7839</td>
<td>KING</td>
<td>none</td>
<td>17-NOV-81</td>
<td>5000</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN</td>
<td>7698</td>
<td>08-SEP-81</td>
<td>1500</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
<td></td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK</td>
<td>7788</td>
<td>12-JAN-83</td>
<td>1100</td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
<td></td>
</tr>
<tr>
<td>7900</td>
<td>JAMES</td>
<td>CLERK</td>
<td>7698</td>
<td>03-DEC-81</td>
<td>950</td>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
<td></td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST</td>
<td>7566</td>
<td>03-DEC-81</td>
<td>3000</td>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
<td></td>
</tr>
<tr>
<td>7934</td>
<td>MILLER</td>
<td>CLERK</td>
<td>7782</td>
<td>23-JAN-82</td>
<td>1300</td>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
<td></td>
</tr>
</tbody>
</table>

---

**Lock Manager**

- **Lock Acquiring**
- **Lock Releasing**
- **Phase 1**
- **Phase 2**
- **End of transaction**
Formal Treatment of Normalization
Finding Keys (Armstrong’s Axioms)

Armstrong’s Axioms:

- if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ (reflexivity) (e.g., if $\alpha = AB$ then $AB \rightarrow A$ and $AB \rightarrow B$)
  (these are the trivial dependencies)
- if $\alpha \rightarrow \beta$, then $\gamma \alpha \rightarrow \beta$ (augmentation)
  and $\gamma \alpha \rightarrow \gamma \beta$
- if $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$ (transitivity)
Finding Keys (additional rules)

Additional rules implied by Armstrong’s Axioms:

- if $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta \gamma$ holds (union)
- if $\alpha \rightarrow \beta \gamma$ holds, then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds (decomposition)
- if $\alpha \rightarrow \beta$ holds and $\gamma \beta \rightarrow \delta$ holds, then $\alpha \gamma \rightarrow \delta$ holds (pseudotransitivity)
Procedure for Computing $F^+$

Finding Keys ($F^+$)

To compute the closure of a set of functional dependencies $F$:

$$F^+ = F$$

repeat

for each functional dependency $f$ in $F^+$

apply reflexivity and augmentation rules on $f$

add the resulting functional dependencies to $F^+$

for each pair of functional dependencies $f_1$ and $f_2$ in $F^+$

if $f_1$ and $f_2$ can be combined using transitivity

then add the resulting functional dependency to $F^+$

until $F^+$ does not change any further
Example

\[ \text{Example} \]

\[ R = (A, B, C, G, H, I) \]

\[ F = \{ \quad A \rightarrow B \]

\[ \quad A \rightarrow C \]

\[ \quad CG \rightarrow H \]

\[ \quad CG \rightarrow I \]

\[ \quad B \rightarrow H \} \]

some members of \( F^+ \)

\[ A \rightarrow H \]

by transitivity from \( A \rightarrow B \) \text{ and } \( B \rightarrow H \)

\[ AG \rightarrow I \]

by augmenting \( A \rightarrow C \) with \( G \), to get \( AG \rightarrow CG \)

and then transitivity with \( CG \rightarrow I \)

\[ CG \rightarrow HI \]

from \( CG \rightarrow H \) \text{ and } \( CG \rightarrow I \) : “union rule” can be inferred from

– definition of functional dependencies, or
– Augmentation of \( CG \rightarrow I \) to infer \( CG \rightarrow CGI \), augmentation of

\[ CG \rightarrow H \] to infer \( CGI \rightarrow HI \), and then transitivity
Finding Keys ($a^+$)

- Closure of a set of attributes:
  - Given a set of attributes “$a$”, define the closure of “$a$” under $F$ (denoted by $a^+$) as the set of attributes that are functionally determined by “$a$” under $F$:
    
    $$\text{if } a \rightarrow \beta \text{ is in } F^+ \text{ then } \beta \subseteq a^+$$

  - Algorithm to compute $a^+$, the closure of $a$ under $F$

    $$\text{result} := a;$$
    $$\text{while (changes to result) do }$$
    $$\text{for each } \beta \rightarrow \gamma \text{ in } F \text{ do}$$
    $$\text{begin}$$
    $$\text{if } \beta \subseteq \text{result then result} := \text{result} \cup \gamma$$
    $$\text{end}$$
Finding Keys

• $\alpha$ is a superkey key if $\alpha^+$ contains all attributes of $R$. It is also a candidate key if no subset of $\alpha$ is a candidate key.
Finding Keys - Example

- $F = \{ A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H \}$

After trying $A^+, B^+, C^+, G^+, H^+, I^+, (AB)^+, (AC)^+$ with no success, $(AG)^+ \rightarrow R$

1. $result = AG$
2. $result = ABCG$ (from $A \rightarrow C$ and $A \rightarrow B$)
3. $result = ABCGH$ (from $CG \rightarrow H$)
4. $result = ABCGHI$ ($CG \rightarrow I$)

AG a superkey because:

$AG \rightarrow R$

AG is a candidate key because:

$A^+$ does not $\rightarrow R$ and $G^+$ does not $\rightarrow R$
Lossless-Join Decomposition

Example of Non Lossless-Join Decomposition

Decompose empdept (create table empdept as select * from emp natural join dept) into 2 tables:

create table emp_dept1 as
    select empno, ename, job, mgr, hiredate, sal, comm from emp_dept
create table emp_dept2 as select job, deptno, dname, loc from emp_dept

Then try to recreate empdept by issuing the following query

select * from emp_dept1 natural join emp_dept2
Lossless-Join Decomposition

Test for Lossless-Join Decomposition

Let R be a relation that you want to decompose into R1 and R2. The decomposition is Lossless if

\[ R_1 \cap R_2 \rightarrow R_1 \text{ or } R_1 \cap R_2 \rightarrow R_2 \]

In other words, if \( R_1 \cap R_2 \) forms a superkey of either R1 or R2.
Dependency Preservation Decomposition

**Example of Non Dependency Preservation Decomposition**

Decompose empdept (create table empdept as select * from emp natural join dept) into 2 tables:

- `create table empdept3 as select empno, ename, job, mgr, hiredate, sal, comm, deptno from empdept`
- `create table empdept4 as select distinct empno, dname, loc from empdept`

No Lossless-Join Problem but a test for the FDs deptno → dname and deptno → loc requires a join.
Dependency Preservation Decomposition

Test for Dependency Preservation Decomposition
Let R be a relation that you want to decompose into R1, R2 … Rn and Let each relation have the following Functional Dependencies R has F, R1 has F1, R2 has F2 and Rn has Fn

The decomposition has Dependency Preservation if

\[(F_1 \cup F_2 \cup … \cup F_n)^+ = F^+\]

For our example F is empno \(\rightarrow\) R and deptno \(\rightarrow\) deptno, dname, loc
F1 is empno \(\rightarrow\) R1
F2 is empno \(\rightarrow\) R2

\[(F_1 \cup F_2)^+\] does not contain deptno \(\rightarrow\) deptno, dname, loc
2NF - Relation R is in 2NF if:

R is in 1NF and

- no nonprime attribute (i.e., an attribute that’s not part of any candidate key) is partially dependent on any candidate key

- (no nonprime attribute is functionally determined by a prime attribute)

- in other words, each nonprime attribute in relation R is fully dependent upon every candidate key

Note: If relation R has no compound keys, R is automatically in 2NF
Designing Databases Using Normalization

2NF – A good way to remember this.

If you have a SINGLE table that is a mapping from the following object model (instead of the normal three tables),

the single table will not be in 2NF.
The problem is that some non-key attributes are functionally dependent on PART of the key (this is not allowed for 2NF).

**Decomposition into 2NF**

A \(\rightarrow\) B \(\rightarrow\) C \(\rightarrow\) D

A \(\rightarrow\) D

B \(\rightarrow\) C

This will be a Lossless-Join decomposition.
3NF - Relation R is in 3NF if:

R is in 2NF and

No nonprime attribute functionally determines any other nonprime attribute
3NF – A good way to remember this.

If you have a SINGLE table that is a mapping from the following object model (instead of the normal two tables),

the single table will not be in 3NF.
The problem is that some non-key attribute(s) are functionally dependent on a non-key attribute (this is not allowed for 3NF).

**Decomposition into 3NF**

Convert to

\[
\begin{align*}
A & \rightarrow B \ C \\
B & \rightarrow C \\
B & \rightarrow C
\end{align*}
\]

This will be a Lossless-Join and Dependency Preserving decomposition.
Designing Databases Using Normalization

**BCNF** - Relation R is in BCNF if:

R is in 1NF and

- All functional dependencies must either be
  - trivial
  - or, their left hand sides must be a superkeys

Note: This means

- all nonprime attributes must be fully dependent on every key (2NF)
- no nonprime attribute functionally determines any other nonprime attribute (3NF)
- (i.e. all BCNF Relations are in 3NF but not all 3NF Relations are in BCNF)
Designing Databases Using Normalization

BCNF Decomposition Algorithm

\( result := \{R\} \);
\( done := \text{false} \);
compute \( F^+ \);

\textbf{while (not done) do}

\textbf{if} (there is a \( R_i \) in \( result \) that is not in BCNF)

\textbf{then begin}

let \( \alpha \rightarrow \beta \) be a nontrivial functional dependency that holds on \( R_i \)
such that \( \alpha \rightarrow R_i \) is not in \( F^+ \) (i.e., \( \alpha \) is not a super key),
and \( \alpha \cap \beta = \emptyset \);
\( result := \{ (result - R_i) \cup (R_i - \beta) \} \cup \{ \alpha \beta \} \)

\textbf{end}

\textbf{else} \( done := \text{true} \);

\textbf{Note: each} \( R_i \) \textbf{is in BCNF, and decomposition is Lossless-Join but not always Dependency Preserving}
A Dependency Diagram
Second Normal Form (2NF) Conversion Results

Table name: PROJECT

Table name: EMPLOYEE

transitive dependency

Table name: ASSIGN

Dr. Philip Cannata
Third Normal Form (3NF) Conversion Results

Table name: PROJECT

Table name: EMPLOYEE

Table name: JOB

Table name: ASSIGN
A Table That is in 3NF but not in BCNF
Decomposition to BCNF

3NF, but not BCNF

1NF

Partial dependency

3NF and BCNF

3NF and BCNF
Sample Data for a BCNF Conversion

<table>
<thead>
<tr>
<th>STU_ID</th>
<th>STAFF_ID</th>
<th>CLASS_CODE</th>
<th>ENROLL_GRADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>25</td>
<td>21334</td>
<td>A</td>
</tr>
<tr>
<td>125</td>
<td>20</td>
<td>32456</td>
<td>C</td>
</tr>
<tr>
<td>135</td>
<td>20</td>
<td>28458</td>
<td>B</td>
</tr>
<tr>
<td>144</td>
<td>25</td>
<td>27563</td>
<td>C</td>
</tr>
<tr>
<td>144</td>
<td>20</td>
<td>32456</td>
<td>B</td>
</tr>
</tbody>
</table>
Another BCNF Decomposition
Fourth Normal Form (4NF)

Table is in fourth normal form (4NF) when both of the following are true:

- It is in 3NF
- Has no multiple sets of multivalued dependencies

4NF is largely academic if tables conform to following two rules:

- All attributes must be dependent on primary key, but independent of each other
- No row contains two or more multivalued facts about an entity
Fourth Normal Form (4NF) (continued)

Database name: Ch05_Service

**Table name: VOLUNTEER_V1**

<table>
<thead>
<tr>
<th>EMP_NUM</th>
<th>ORG_CODE</th>
<th>ASSIGN_NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>10123</td>
<td>RC</td>
<td>1</td>
</tr>
<tr>
<td>10123</td>
<td>UW</td>
<td>3</td>
</tr>
<tr>
<td>10123</td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

**Table name: VOLUNTEER_V2**

<table>
<thead>
<tr>
<th>EMP_NUM</th>
<th>ORG_CODE</th>
<th>ASSIGN_NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>10123</td>
<td>RC</td>
<td></td>
</tr>
<tr>
<td>10123</td>
<td>UW</td>
<td>1</td>
</tr>
<tr>
<td>10123</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>10223</td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

**Table name: VOLUNTEER_V3**

<table>
<thead>
<tr>
<th>EMP_NUM</th>
<th>ORG_CODE</th>
<th>ASSIGN_NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>10123</td>
<td>RC</td>
<td>1</td>
</tr>
<tr>
<td>10123</td>
<td>RC</td>
<td>3</td>
</tr>
<tr>
<td>10123</td>
<td>UW</td>
<td>4</td>
</tr>
</tbody>
</table>
Fourth Normal Form (4NF) (continued)