## Normalization:

Normalization is a process to eliminate redundancy, as redundancy is the root cause of a number of problems.

Problems brought about by redundancy:

a) Insertion anomalies
b) Deletion anomalies
c) Updation anomalies.

Eg: Student

| Student ID | Name | Age | PrgID | PrgName | PrgChair |
|---|---|---|---|---|---|
| 101 | A | 20 | 111 | CSC | XYZ |
| 102 | B | 18 | 111 | CSC | XYZ |
| 301 | L | 21 | 111A | CSE | XYZ |

These anomalies are brought by storing program info in one student table.

↳ This can be resolved if we adopt the ER idea.

Student

| Student ID | Name | Age | PrgID (FK) |
|---|---|---|---|
| 101 | A | 20 | 111 |
| 102 | B | 18 | 111 |
| 301 | L | 21 | 111A |

| PrgID | PrgName | PrgCh |
|---|---|---|
| 111 | CSC | XYZ |
| 111A | CSE | XYZ |

This logical break down is brought about by Normalization

1NF → 2NF → 3NF → BCNF {using Functional Dep's}

## 8 Functional Dependencies (FD):
(this is a tool/technique)

Consider the relation $R(\alpha, \beta)$ and the functional dependency $fd: \alpha \rightarrow \beta$

| R | $\alpha$ | $\beta$ |
|---|---|---|
| | a | 1 |
| | b | 1 |
| | c | 3 |
| | d | 4 |

determinant → dependent

$fd: \alpha \rightarrow \beta$

↑ search function

i.e., $\rightarrow \approx$ search function; what does this mean?
consider a function $f(x) = y$ and we make observations on 'y' by changing the values of 'x'.

$f(1) = a$
$f(2) = b$
$f(1) = c$

} do you think the function is correct?

the answer is "no"! {this is because when $x = 1$, we observe different values of $y$}

So, bringing this idea back to a search function/fd; A fd is a function that enables you to find the value of $\beta$ given $\alpha$; and this is possible only when the values of $\alpha$ preserves a logical ordering in a relation.

so if we go back to relation R; the following edits are made.

| R | $\alpha$ | $\beta$ |
|---|---|---|
| $t_1 \rightarrow$ | a | 1 |
| $t_2 \rightarrow$ | a | 1 |
| | c | 3 |
| | d | 4 |

$\alpha \subseteq R$, $\beta \subseteq R$

if $\alpha \rightarrow \beta$ "holds true" then
$t_1[\alpha] = t_2[\alpha]$ implies
$t_1[\beta] = t_2[\beta]$.

## Types of FD's:

a) trivial: $AB \rightarrow A$ i.e. here $\beta \subseteq \alpha$

b) non-trivial: $AB \rightarrow CBA$ i.e. here $\beta \nsubseteq \alpha$ / $\alpha \subseteq \beta$

c) completely non-trivial: $AB \rightarrow DE$ i.e. $\alpha \cap \beta = \phi$

Example:

| R | A | B | C | D | E |
|---|---|---|---|---|---|
| | a | 2 | 3 | 4 | 5 |
| | 2 | a | 3 | 4 | 5 |
| | a | 2 | 3 | 6 | 5 |
| | a | 2 | 3 | 6 | 6 |

a) $A \rightarrow BC$

b) $DE \rightarrow C$

c) $C \rightarrow DE$

d) $BC \rightarrow A$

Given R, determine if a, b, c, & d are valid functional dependencies.

{HINT: look for a given value of x, do you get the same value of $\beta$.}

## Important Examples:

$R_2$

| A | B | C | D | E |
|---|---|---|---|---|
| a | 2 | 3 | 4 | 5 |
| b | a | 3 | 4 | 5 |
| c | 2 | 3 | 6 | 5 |
| d | 2 | 3 | 6 | 6 |

Set 1:

a) $A \rightarrow BCD$
b) $A \rightarrow DE$
c) $A \rightarrow E$

Set 2:

a) $AB \rightarrow C$
b) $ED \rightarrow C$
c) $EA \rightarrow C$

- Observation: SET1: all values of A are unique ∴ all the fd's are valid; i.e all values of the "$\alpha$" (determinant) are unique

SET2: all fd's are valid if the values of "$\beta$" (dependants) are identical.

$R_3$

| X | Y | Z |
|---|---|---|
| 1 | 4 | 2 |
| 1 | 5 | 3 |
| 1 | 6 | 3 |
| 3 | 2 | 2 |

a) $XY \rightarrow Z$ && $Z \rightarrow Y$
b) $YZ \rightarrow X$ && $Y \rightarrow Z$
c) $YZ \rightarrow X$ && $X \rightarrow Z$
d) $XZ \rightarrow Y$ && $Y \rightarrow Z$

$R_4$

| A | B | C |
|---|---|---|
| 1 | 2 | 4 |
| 3 | 5 | 4 |
| 3 | 7 | 2 |
| 1 | 4 | 2 |

a) $A \rightarrow B$ && $BC \rightarrow A$
b) $C \rightarrow B$ && $CA \rightarrow B$
c) $B \rightarrow C$ && $AB \rightarrow C$
d) $A \rightarrow C$ && $BC \rightarrow A$

## 8 Hidden Functional Dependencies:

Library Database: BOOK (ACC_NO, YR_PUB, TITLE)
USER (CARD_NO, B_NAME, B_ADDR)
SUPPLIER (S_NAME, S_ADDR)
BORROW (ACC_NO, CARD_NO, DOI)
SUPPLY (ACC_NO, S_NAME, PRICE, DOS)

fd: S_NAME, DOS → PRICE (supply table)
CARD_NO → DOI (borrow table/schema)
ACC_NO → YR_PUB
ACC_NO, TITLE → YR_PUB
ACC_NO → YR_PUB, TITLE
} (Book schema) too many can we reduce

Step 1: Look for redundancies in K & B
ACC_NO, TITLE → YR_PUB
ACC_NO → YR_PUB
ACC_NO → YR_PUB, TITLE ← redundant (not required)

step 2: Weed out hidden f.d's.
ACC_NO → S_NAME (supply)
ACC_NO → S_ADDR ← redundant (there is a hidden dependancy)
S_NAME → S_ADDR (supplier)

i.e. ACC_NO → S_ADDR can be infered using
ACC_NO → S_NAME
S_NAME → S_ADDR.