

# Database relazionali: normalizzazione

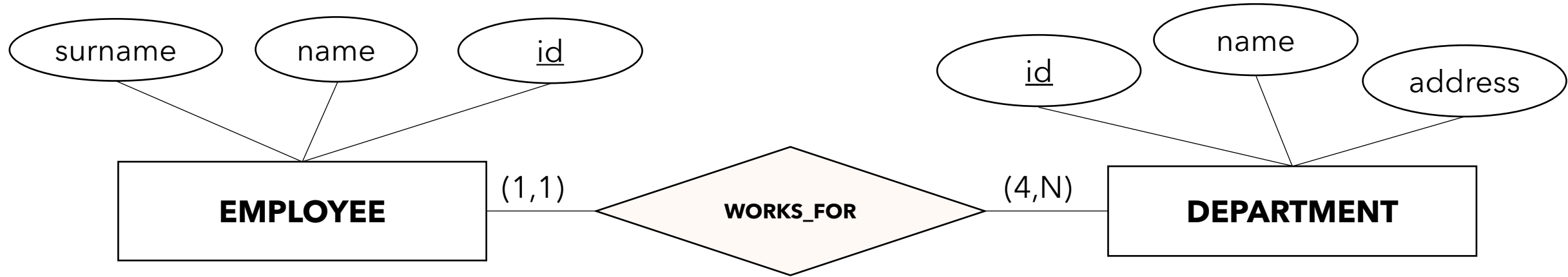
**Liceo G.B. Brocchi - Bassano del Grappa (VI)**  
**Liceo Scientifico - opzione scienze applicate**  
Giovanni Mazzocchin

# Normalizzazione di uno schema relazionale

Partiamo dalle considerazioni seguenti:

1. una relazione di uno schema relazionale dovrebbe derivare da una singola entità dello schema concettuale
2. uno schema relazionale dovrebbe essere semplice da spiegare in linguaggio naturale

# Normalizzazione di uno schema relazionale



`emp_dept(emp_id, emp_name, emp_surname, dept_id, dept_name, dept_address)`

Difetto di progettazione: 2 entità diverse tradotte in un'unica relazione

# Update anomalies

- Lo schema logico precedente *soffre* di una **insertion anomaly** (*anomalia di inserimento*):
  - per inserire un dipendente, dovremmo specificare anche tutte le informazioni relative al reparto per cui lavora
  - se non lavora per nessun reparto, dovremmo inserire valori NULL per dept\_id, dept\_name e dept\_address
  - e se volessimo archiviare le informazioni relative ad un reparto che non ha ancora dipendenti?

# Update anomalies

- Lo schema logico precedente *soffre* di una **deletion anomaly** (*anomalia di cancellazione*):
  - se cancelliamo il record dell'ultimo dipendente rimasto in un reparto, scompariranno anche le informazioni relative al reparto

# Update anomalies

- Lo schema logico precedente *soffre* di una **modification anomaly** (*anomalia di modifica*):
  - se la location di un reparto cambia, dobbiamo aggiornare tutte le tuple dei dipendenti che lavorano in quel reparto
  - se ci dimentichiamo di aggiornare anche solo una tupla, il database sarà in uno stato *inconsistente*

# Dipendenze funzionali

- **Definizione:** una dipendenza funzionale  $X \rightarrow Y$ , dove  $X$  e  $Y$  sono due insiemi di attributi di una relazione  $R$ , è il seguente vincolo:
  - per ogni coppia di tuple  $t_1$  e  $t_2$  di  $R$  tali che  $t_1[X] = t_2[X]$ , vale anche  $t_1[Y] = t_2[Y]$
- Esempio: relazione  $R(A, B, C, D)$ , con una sua estensione

A	B	C	D
a1	b4	c1	d1
a2	b4	c1	d2
a3	b1	c2	d3
a1	b4	c3	d4

# Dipendenze funzionali

- **Definizione:** una dipendenza funzionale  $X \rightarrow Y$ , dove  $X$  e  $Y$  sono due insiemi di attributi di una relazione  $R$ , è il seguente vincolo:
  - per ogni coppia di tuple  $t_1$  e  $t_2$  di  $R$  tali che  $t_1[X] = t_2[X]$ , vale anche  $t_1[Y] = t_2[Y]$
- **NB:** una FD (*functional dependency*)  $X \rightarrow Y$  sussiste se non è violata
- Esempio: relazione  $R(A, B, C, D)$ , con una sua estensione

A	B	C	D
a1	b4	c1	d1
a2	b4	c1	d2
a3	b1	c2	d3
a1	b4	c3	d4

$A \rightarrow B?$   
 $A \rightarrow C?$   
 $B \rightarrow C?$   
 $B \rightarrow D?$   
 $C \rightarrow D?$   
 $D \rightarrow C?$   
 $D \rightarrow B?$   
 $\{B, C\} \rightarrow A?$



# Normalizzazione

- Gli obiettivi della **normalizzazione** sono:
  1. la minimizzazione delle ridondanze
  2. la minimizzazione delle anomalie
- Uno schema relazionale che non rispetta una forma normale verrà quindi scomposto in più schemi relazionali (conservando però le stesse informazioni)

# First Normal Form (1NF)

- La **1NF** richiede che in una relazione i valori degli attributi siano tutti **atomici**

DEPARTMENT		
<u>id</u>	name	locations
1	IT	San Francisco, Cupertino, Los Angeles
2	Sales	San Diego
3	HR	San Diego, Sacramento

**la relazione DEPARTMENT viola la 1NF: i valori di locations non sono tutti atomici**

# First Normal Form (1NF)

- La **1NF** richiede che in una relazione i valori degli attributi siano tutti **atomici**

DEPARTMENT	
<u>id</u>	name
1	IT
2	Sales
3	HR

DEPT_LOCS	
<u>id_dept</u>	location
1	San Francisco
1	Cupertino
1	Los Angeles
2	San Diego
3	San Diego
3	Sacramento

- questa progettazione logica rispetta la 1NF
- le informazioni della versione non normalizzata sono conservate

# Second Normal Form (2NF)

- La **2NF** richiede che in una relazione R tutti gli attributi non parte della PK siano completamente funzionalmente dipendenti dalla PK
- $X \rightarrow Y$  è una **full functional dependency** se  $X - \{A\} \rightarrow Y$  non è più una functional dependency

doct_pat			
<u>doct_ssn</u>	<u>pat_ssn</u>	doct_name	pat_name
1	2	giovanni	maria
1	3	giovanni	mario
2	5	giuseppe	francesco

non è in 2NF, infatti:  
 $\text{doct\_ssn} \rightarrow \text{doct\_name}$   
 $\text{pat\_ssn} \rightarrow \text{pat\_name}$

la PK è {doct\_ssn, pat\_ssn},  
ma gli altri attributi dipendono da  
sottoinsiemi della PK

# Third Normal Form (3NF)

- Uno schema relazionale R è in **3NF** se rispetta la 2NF, e nessun attributo non primo di R è transitivamente funzionalmente dipendente dalla PK
- Una FD  $X \rightarrow Y$  è **transitiva** se esiste un insieme di attributi Z non primi tale che valgono le FD  $X \rightarrow Z$  e  $Z \rightarrow Y$

DOCT_HSP_WARD				
<u>d_ssn</u>	d_name	w_number	w_name	w_head_ssn
1	mario	3	cardiology	4
2	maria	3	cardiology	4
3	giuseppe	4	ER	6
4	francesca	4	ER	6

## Dipendenze funzionali:

$d\_ssn \rightarrow w\_head\_ssn$  **sì**

ma anche:

$d\_ssn \rightarrow w\_number$  **sì**

$w\_number \rightarrow w\_head\_ssn$  **sì**

X: d\_ssn, Y: w\_head\_ssn,

Z: w\_number

DOCT\_HSP\_WARD viola la 3NF