



L3 - Basi di Modellazione

Created	@September 23, 2022
Class	Basi di Dati
Materials	<u>L03_1.pdf</u>

Ciclo di vita dei sistemi informativi

1. Studio di fattibilità → Si analizzano le potenziali aree di applicazione, si effettuano degli studi di costi/benefici, si determina la complessità di dati e processi, e si impostano le priorità tra le applicazioni
2. Raccolta e analisi dei requisiti → Comprende una raccolta dettagliata dei requisiti con interviste ai potenziali utenti, per definire le funzionalità del sistema
3. Progettazione dei dati
4. Realizzazione → Si implementa il sistema informativo, si carica il DB e si implementano e si testano le transazioni
5. Validazione e collaudo → si verifica che il sistema soddisfi i requisiti e le performance richieste
6. Funzionamento → La fase operativa del nuovo sistema parte quando tutte le funzionalità sono state validate

Le fasi di raccolta dei requisiti e di progettazione di dati sono intrinsecamente legate alla creazione della base di dati

Abbiamo due tipi principali di modelli di dati:

- Modelli concettuali → Rappresentano i dati in modo indipendente da ogni sistema. Cercano di descrivere i concetti del mondo reale e sono usati nelle fasi preliminari
 - Si utilizza per astrarre la fase di pianificazione e non considerare dati specifici che non sono richiesti

- Modelli logici → Si usano per l'organizzazione dei dati e sono indipendenti dalle strutture fisiche
-

Progettazione concettuale

- Individuare l'informazione rilevante da memorizzare per soddisfare le richieste informative e funzionali
- Interesse specifico per la parte statica del problema: i dati
- Individuazione nel problema degli aspetti informativi rilevanti:
 - Entità → Gli elementi informativi
 - Relazioni → Correlazioni tra gli elementi
 - Vincoli → Vincoli per la coerenza delle informazioni
 - Interrogazioni → Interrogazioni dagli utenti al database
 - Frequenza di iterazioni
 - Numero di utenti che interagiscono in contemporanea

I diagrammi usati principalmente sono i diagrammi ER, insieme ai class diagram di UML

- ER → Entity Relationship → Definito per le basi di dati relazionali con costrutti e forme adatte alla modellazione di un database
 - UML → Unified Modelling Language → Sono utilizzate per molte attività di modellazione nel ciclo di vita di un progetto software
-

Modellazione concettuale

MODELLO ER:

Il modello ER è diffusissimo data model di alto livello, utilizzato per definire lo schema concettuale di un database. È stato concepito per essere più vicino ai concetti "umani", e quindi facilmente comprensibile anche ad utenti non tecnici

Il modello ER descrive i dati con tre concetti fondamentali:

- Entità
- Attributi
- Relazioni

MODELLO UML:

L'UML è un linguaggio grafico per la modellazione di applicazioni software basate sulla programmazione orientata agli oggetti. Può anche essere utilizzato in alternativa al modello ER, dove i diagrammi delle classi sono usati per descrivere la classi di interesse e le relazioni che intercorrono tra di esse

DIFFERENZE TRA GLI APPROCCI:

- Assenza di notazione standard per definire gli identificatori
 - Possibilità di aggiungere note per commentare i diagrammi
 - Possibilità di indicare il verso di navigazione di un'associazione
 - Formalismi diversi
-

Esempio di database: DB Company

REQUISITI:

- La compagnia è organizzata in DIPARTIMENTI. Ogni dipartimento ha un nome, un numero e un impiegato che lo gestisce. Bisogna tener traccia della data di insediamento del manager. Un dipartimento può aver più locazioni
 - Ogni dipartimento controlla una serie di PROGETTI. Ogni progetto ha un nome, un numero e una singola locazione
 - Per IMPIEGATO si tiene traccia di: nome, SSN, indirizzo, salario, genere e data di nascita. Ogni impiegato lavora per un dipartimento e può lavorare su più progetti. Teniamo traccia del numero di ore settimanali che un impiegato spende su un progetto e del supervisore di ogni impiegato
 - Ogni impiegato ha una serie di PERSONE A CARICO. Per ogni persona a carico, registriamo: nome, sesso, data di nascita e parentela con l'impiegato
-

Entità e attributi

ENTITÀ:

Le entità corrispondono a classi di oggetti del mondo reale che hanno proprietà omogenee, comuni ed esistenza autonomi ai fini dell'applicazione di interesse. Un'entità E può essere un oggetto fisico o un oggetto concettuale

ATTRIBUTI:

Ogni entità ha delle proprietà dette attributi. Ogni entità è caratterizzata da un set di valori per i suoi attributi

- Nel diagramma ER un attributo descrive una proprietà elementare di un'entità o di una relazione
- Nel class diagram un attributo rappresenta una proprietà elementare degli oggetti di una classe

Gli attributi possono essere di più tipi:

- Attributi semplici → Ogni entità ha un valore singolo (atomico) per tale attributo
- Attributi composti → Possono essere divisi in sottoparti, che rappresentano informazioni di base con loro significati
- Attributi Singlevalued → Hanno un solo valore per ciascuna entità
- Attributi Multivalued → Può avere un insieme di valori per la stessa entità. In UML può essere determinato un limite inferiore e superiore al numero di valori per un'entità
- Attributi Archiviati e derivati → Alcuni attributi possono essere in relazione fra di loro, e da uno può essere calcolato l'altro (es: data di nascita, età)
- Attributi Complessi → In alcuni casi gli attributi composti e gli attributi multivalued possono essere annidati in modo arbitrario

Attributi chiave

Un importante vincolo sulle entità è la chiave, o vincolo di unicità. L'attributo chiave deve avere un valore univoco per ogni entità e serve per identificarle univocamente. Talvolta una può essere formata da un attributo composto