INTRO

Azienda

**Organizzazione (pubblica o privata) strutturata e organizzata secondo lo scopo per cui è nata (mission) e per perseguire degli obiettivi generali (target)**.

Risorsa

**Tutto ciò con cui un'azienda opera per raggiungere i propri target e la propria mission** (da **materiali** a **persone**)

Processo

**Insieme delle attività** (decisioni e azioni) **che l'azienda svolge per gestire il ciclo di vita di 1 o + risorse**.

Processo software house --> analisi --> progettazione --> sviluppo --> collaudo --> installazione

Waterfall

Analisi --> Progettazione --> Realizzazione --> Collaudo --> Manutenzione

Informazione

La **risorsa più importante,** riguarda tutte le altre in un'azienda. È importante che venga **condivisa** nell'organizzazione.

Dato (ES: 45. *Che cazzo è 45?*)

È la **descrizione elementare di una cosa**, ma **non** **è** **informazione**. Per esserlo gli serve la **chiave di interpretazione** (**significato**).

Informazione (ES: 45 km. *Ah ecco.*)

È **l’incremento di conoscenza fornito da un dato**. Tutto ciò che aumenta la mia conoscenza è informazione.

TIPI DI SISTEMI

Sistema Informativo

È l'in**sieme di strumenti automatici, procedure manuali, norme organizzative, risorse umane e materiali orientato alla gestione** (raccolta, archiviazione, elaborazione e scambio) **delle informazioni importanti e necessarie alle attività operative, di gestione, di preparazione, controllo e valutazione dell'organizzazione**.

In pratica è **ciò che serve al fine di gestire le informazioni necessarie al buon funzionamento dell'organizzazione**.

Tutte le aziende hanno un sistema informativo (semplice o complesso). Questo è ciò che gestisce le informazioni.

Basic example

Si considera team di vendita di una ditta artigianale diretta da Rossi. Non farà tantissime fatture, e avrà tot clienti che richiedono un lavoro base. Quindi il suo sistema informativo può includere: una rubrica con info personali dei clienti, biglietti da visita, le fatture e gli ordini.

Complex example

Esempio di compagnia aerea, qui serve avere lista e uno storico clienti, lista, storico e orario piloti e hostess, manutenzioni aerei, caratteristiche e lista aerei, prenotazioni, voli e passeggeri, magazzino pezzi ricambio aerei, contabilità, rifornimento e approvvigionamento di kerosene per aerei, internet clienti e privato di compagnia...

Tipi di sistemi informativi

Ufficiale

Ciò che l'azienda crea e predispone per consentire il flusso di informazioni in essa.

Privato

Informazioni che scorrono al di fuori del canale ufficiale.

Tipi di informazioni

Formale

Informazioni strutturate ed analizzate in modo deterministico. (Esempio comunicazione malattia: modulo)

Informale

Informazioni caratterizzate dall'aleatorietà. (Esempio comunicazione malattia: chiamata o messaggio whatsapp)

Livello di aggregazione/conservazione/filtraggio di informazioni

Le **info vanno aggregate in base al livello di impiego**. Il loro grado di aggregazione segue un modello a **piramide**:

Livello strategico

Dei capi e CEO, a loro interessa solo che i team leader consegnino i progetti in tempo.

Livello Tattico

Dei team leader, a loro interessa che i programmatori portino a termine il progetto entro un certo tempo.

Livello Operativo

Dei programmatori, a loro interessa finire le task e risolvere i problemi in esse.

Sistema Informatico

È **l'insieme degli strumenti informatici usati per il trattamento automatico delle informazioni rappresentante con dati digitali per agevolare le funzioni del suo sistema informativo** (di cui è parte).

Detto anche **EDP** (Electronic Data Processing)

**In piccolo** (ditta artigianale): potrebbe essere una rubrica elettronica dove sono salvate le info dei clienti

**In grande** (compagnia aerea): archivi di dati di aerei, personale, clienti, voli... siti web e programmi per prenotazioni...

Applicazione Informatica

**Componente del sistema informatico che vi utilizza i dati per eseguire una funzione nell'azienda.**

ARCHIVI

Archivi

**Raccolta organizzata di informazioni necessarie all'azienda** (*elaborabili in qualsiasi forma, digitali o fisiche*).

Sono **strutture dati astratte** (*solo di struttura, tipo per indicare suddivisioni o sort*), costituite da insiemi di **record**.

Le info vanno memorizzate con una logica tale da rendere la loro consultazione il più efficiente possibile.

Ogni record all'interno dell'archivio è identificato grazie alla sua posizione, ovvero, un **indirizzo logico**.

File

Sono **archivi su memoria di massa**. (**File**: **insieme di registrazioni omogenee memorizzate in memoria di massa**.)

Le **informazioni** sono memorizzate in essi come **sequenze di byte** o **record**. Idoneo quindi ad essere un archivio.

Un **archivio è SEMPRE implementato mediante 1 o + file**. Ma **un file NON è SEMPRE un archivio**. (tipo .exe)

Record

Sono gli **elementi che compongono un file**

Campi

Formano un record. Ciascuno di essi **contiene un'informazione**. (File = tabelle, righe = record, colonne = campi)

I **campi** sono **omogenei**, se sono tutti dello **stesso tipo** e sono nello **stesso ordine**.

Tracciato Record

È **l'articolazione dei campi nel record**.

Definisce:

* Campi,
* Tipo di campi,
* Lunghezza dei campi,
* Operazioni possibili su essi.

|  |  |
| --- | --- |
| Primary Key (PK) | Secondary/Foreign Key (FK) |
| **Campo che identifica univocamente un record.** | **Campo che identifica un record nell'archivio.** |

Tipi di record

Logico

**Struttura che il programmatore** **dà** **ai record del proprio archivio**. Diviso in campi e di dimensione fissa (somma della lunghezza di ciascun campo). Fatta dal **programmatore**.

Fisico (o blocco)

**Insieme dei byte che possono essere letti o scritti in memoria con 1 singola operazione di lettura/scrittura**.

**1 record fisico può contenere + record logici** (in base a come SO organizza lo spazio su disco)

Organizzazione degli archivi

È il **modo in cui sono rappresentati in memoria**. Può essere:

Fisica (*come sono memorizzati su memoria di massa*)

* Supporti ad **accesso sequenziale** (**nastri**)
* Supporti ad **accesso diretto** (**dischi**)

Logica (*dipende dalla fisica*)

* **Sequenziale**
* **Sequenziale ad indici**
* Ad **accesso diretto**

Operazioni sugli archivi

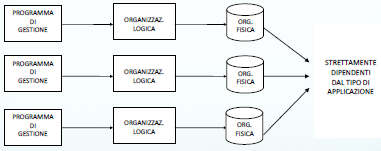
Operazioni su file: apro, scrivo, leggo, appendo, cancello, creo, chiudo... Invece sugli archivi:

* Ricerca
* Interrogazione
* Manipolazione
  + INSERT INTO
  + UPDATE (o ALTER)
  + DELETE
* Operazioni globali
  + SELECT
  + ORDER BY
  + JOIN

Scelta dell'organizzazione dell'archivio (dipende da):

* tipo di supporto fisico su cui sarà memorizzato il file
* tipo di operazioni da effettuare sugli archivi
* tempi e metodi di elaborazione (lenti o veloci)
* linguaggio di programmazione

Limiti degli archivi classici

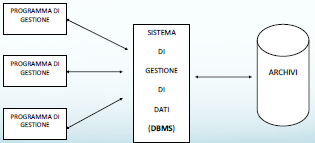
Prima per archivi si usavano file con diverso tipo di accesso, ma c’erano limiti. L’**organizzazione logica** era **diversa**:

1) Gestione ordini

2) Gestione fatture

3) Gestione clienti

Problema

Gestione fatture deve poter accedere a gestione clienti o ordini, ma avendo organizzazioni logiche diverse, non possono farlo se non a mano.

Soluzione

1) **1 unico contenitore di dati** (archivio, poi base di dati)

2) **1 unico programma di gestione archivio o DBMS**

DATABASE

Base di dati

**Raccolta di dati logicamente correlati, usata per modellare una realtà**.

Dati memorizzati su memoria di massa e progettati per essere fruiti in maniera ottimizzata da differenti programmi e utenti.

SQL (Structured Query Language)

**Linguaggio di interrogazione usato dai DBMS relazionali**. I db relazionali sono anche detti **ACID**.

ACID

Atomico

Transazioni sono atomiche, ovvero il DBMS deve essere in grado di annullare tutti i cambiamenti se fallisce.

Consistente

Dati memorizzati nelle tabelle sono coerenti tra loro, e le transazioni non devono violare vincoli di integrità fissati (tipo relazioni con chiavi esterne).

Integro

Non ci sono dati "*fake*" (non mancano informazioni...). Gli effetti delle transazioni devono essere indipendenti da tutte le altre eseguite in concorrenza (lock)

Durevole

Informazioni sono archiviate in modo durevole nel tempo (per sempre).

NoSQL

**DB che consentono memorizzazione ed esecuzione di query** **non** **su** **tradizionali strutture di un DB relazionale SQL**.

Vantaggi

* **Scalabilità** --> **mantiene performance** **all'aumentare** della quantità **di dati**.
* **Velocità** --> **memorizzazione ed elaborazione più rapide**.

Svantaggi

* **Non sono completamente ACID** (o non lo sono proprio)

Tipi

Key-Value

La forma + semplice di db NoSQL, con uno **modello di dati organizzato in KeyValuePairs**.

**Nel valore** **possono** anche **esserci** dei **dati non correlati tra loro o di tipo diverso**.

Documents

Questi **archiviano i dati come documenti**, generalmente in formati **XML**, **JSON** o **BSON**. (+ usato è **MongoDB**)

Wide Columns (Column-family)

Questi **memorizzano le informazioni in colonne**, **consentendo** agli utenti di **accedere** **solo alle colonne che servono** **per allocare meno memoria** inutile. Cerca di risolvere problemi con gli altri 2 tipi, ma sempre + **complesso**.

Graph

Questo **ospita i dati in un grafo di conoscenza**, **memorizzando gli elementi come** **nodi**, **archi** e **proprietà**.

**Nodi** = oggetti/persone

**Archi** = definiscono relazione tra nodi

DBMS

DBMS (Database Management System)

**Software che si occupano della gestione dei database, quindi di memorizzazione e organizzazione dei dati**.

Insieme di programmi che gestiscono i dati (database) lasciando il loro uso ai programmi.

1) Con archivi classici, i programmi dovevano interfacciarsi con il file system, mentre ora è il DBMS che lo usa, quindi si colloca tra i file di dati (database) e i programmi.

2) Quando i programmi necessitano dati, li richiedono al DBMS, che li preleva dalla base di dati col file system e li restituisce ai richiedenti.

DBMS più usati: **Oracle**, **SQL Server**, **MySQL**.

Funzionamento

Gestione DB

Permette operazioni di CREATE, INSERT, UPDATE e SELECT (...) tramite interfacce semplici e intuitive

Consistenza

Garantisce la validità delle info nel db

Integrità dei dati

Garantisce l’integrità dei dati con vincoli di consistenza e validazione input.

Eliminazione ridondanze

Evita la ridondanza (e la controlla se serve), dato che nel db non ci possono essere campi = con valori o archivi diversi.

Sicurezza e protezione dati

Garantisce privatezza di dati con password e crittografia delle info

Supporto di transazioni

Transazione

**Sequenza di operazioni effettuate su un database che può concludersi con successo** (quindi apporta le modifiche al database) **o** **insuccesso** (il database rimane come prima)

Protezione guasti e ripristino

Effettua backups, memorizza transazioni e ha meccanismi di disaster recovery

Accesso concorrente

Garantisce accesso e aggiornamenti concorrenti (+ utenti per volta), impedendo inconsistenze nel db e interferenze.

Gestione dizionario dei dati

Contiene **metadati**, ovvero info che descrivono gli oggetti del db (nomi tabelle attributi, relazioni, vincoli, auth...)

Indipendenza dei dati

Nasconde la reale organizzazione dei dati nei supporti fisici di memoria, grazie a architettura a 3 livelli:

* **View** (o sottoschemi),
* **Schema logico**,
* **Schema fisico**.

E garantisce una visione astratta del db (uso di + utenti)

Livelli di astrazione di un DBMS

Sono 3, e permettono al DBMS di nascondere la reale organizzazione dei dati nei supporti fisici di memoria.

Esterno (Applicativo)

È **come i dati vengono mostrati agli utenti** (sempre mantenendo la loro organizzazione logica e fisica, ma mostrati secondo il formato richiesto). Sono gli stessi utenti che a seconda dell'uso modificano la visualizzazione dei dati.

Descrive i dati come sono visti da applicazioni o utenti (e fornisce strumenti per modificare e vedere i dati, quindi ci operano i singoli utenti del db)

Logico

È **come i dati sono organizzati e correlati logicamente nel database**. Comprende **tabelle**, **chiavi**, **relazioni**, **campi**, **ecc**. Il **programmatore** si occupa di questi e **li gestisce**.

Descrive gli oggetti di interesse per programmi, offrendo rappresentazione precisa dell'organizzazione dei dati (...)

Fisico (Interno)

È **come i dati sono memorizzati sul supporto fisico** (memoria di massa), o collezione di files nella memoria di massa. Questo è **gestito dal DBA** (Database Administrator). Descrive strutture fisiche di archivi che formano il database.

View (vista)

**Visione astratta di una parte del database creata apposta** (su misura) **per l'utente**. Queste permettono di presentare in modi diversi i dati, dando una visualizzazione personalizzata del database.

Indipendenza dei dati (Permessa al DBMS dall'organizzazione a 3 livelli)

Fisica

Con questa i **programmi** sono **indipendenti dai dati fisici**, ovvero, è possibile **cambiare organizzazione fisica dei dati** **senza modificare i programmi** stessi e le procedure che agiscono sul database. (*Tipo se* *posto db da un disco a un altro*)

Logica

Con questa **si può modificare l'organizzazione logica** (*schema*) **dei dati o del database** **senza** **dover** **modificare** **i** **programmi** interessati alla modifica.

DB SPECIFICS

Linguaggio db

(In generale detto SQL) ma si divide in sottolinguaggi:

A livello fisico

**DMCL** (Device Media Control Language)

Comandi o istruzioni che mi permettono di **creare fisicamente file**, **strutture**, **dati**, **formati**, **memorizzazione**, **metodi** **di accesso**... usato dal **DB Administrator**

A livello logico

**DCL** (Data Control Language)

Quello che **gestisce e permette gli accessi al database** (+ visibilità a certi dati)

**DDL** (Data Definition Language)

Linguaggio che permette di **creare le tabelle e relazioni**. Usati da **programmatori**.

A livello esterno

**DML** (Data Manipolation Language)

Linguaggio che permette di **modificare i dati del DB**.

**QL** (Query Language)

Permette di **interrogare il DB con ricerche**.

Utenti

DBA (Database Administrator)

Gli è **affidata la responsabilità di gestione del database**. Compiti del DBA:

* Gestione e trattamento dati,
* Definizione view,
* Manutenzione DB,
* Controllo disponibilità su memoria di massa,
* Implementazione del DB fisico.

Programmatori

**Creano programmi per utilizzare i dati del DB.**

Utenti avanzati

**Accedono al DB per interrogazione o modifica con SQL.**

Utenti semplici

**Interrogano il database con interfacce già fatte** (magari da programmatori).

Chiave di interpretazione in DBMS vs archivi

Gli **archivi classici** **hanno la chiave di interpretazione** **nel tracciato record** (**esterna**)

I **DBMS** **hanno la chiave di interpretazione** **inclusa nel database**, grazie al **dizionario dei dati** (**interna**)

Schema (colonne)

È la **chiave di interpretazione dei dati**. Il significato dato al dato per ricavare l'informazione. Lo schema è **statico.**

Istanza (riga di tabella) (di schema o estensione)

**L'insieme dei valori assunti dallo schema in un** certo **istante di tempo**. Le istanze sono **dinamiche.**

Categoria (nome di tabella)

**Insieme di dati con la stessa chiave di interpretazione**.

Occorrenza

**Insieme delle istanze delle categorie in un istante di tempo** (in numero?)

Esempio:

**Nome tabella** = articoli = **CATEGORIA**

**Nomi colonne** (+ tipo, dimens...) = codice art, qta = **SCHEMA**

**Dati** = (100, 23), (101, 12) = **ISTANZE**