Appunti informatica

## Definizioni base:

* **Basi di dati:** archivi di dati organizzati in modo integrato attraverso tecniche di modellazione dei dati e gestiti sulle memorie di massa dei computer attraverso appositi software.
* **Database:** è una collezione di archivi di dati ben organizzati e ben strutturati per costituire una base di lavoro per utenti diversi con programmi diversi.
* **Consistenza:** degli archivi, i dati contenuti devono essere significativi ed essere utilizzabili nelle applicazioni dell’azienda. **Sicurezza** significa impedire che il database venga danneggiato da interventi accidentali o non autorizzati, **integrità** significa garantire che le operazioni effettuate sul database da utenti autorizzati non provochino una perdita di consistenza ai dati.
* **DBMS:** prodotti software per la gestione di database.
* **Database distribuiti:** gli archivi del database possono risiedere sulle memorie di massa di diversi computer facenti parte di una rete aziendale.
* **Ridondanza:** ripetizione inutile.
* **Incongruenza:** può essere portata dalla ridondanza nel caso un dato venga aggiornato in un archivio e no in un altro.
* **Inconsistenza:** i dati a disposizione non sono più affidabili.

## Modelli di Database

Con la teoria dei database si introduce una nuova metodologia di organizzazione degli archivi, le sue caratteristiche principali: indipendenza dalla struttura fisica dei dati, dalla struttura logica, utilizzo da parte di più utenti, eliminazione della ridondanza e dell’inconsistenza, facilità di accesso, integrità e sicurezza dei dati, linguaggi per la gestione del database.

**Flat file:** database con una struttura di dati formata da un unico file.

**Modello gerarchico:** adatto per rappresentare situazioni nelle quali è possibile fornire all’insieme di dati una struttura nella quale ci sono entità che stanno in alto ed entità che stanno in basso, secondo uno schema ad albero, nel quale i nodi rappresentano le entità e gli archi le associazioni. Può essere un unico record radice con un insieme ordinato di sottoalberi. Questo modello è usato per l’associazione 1:N. Questo sistema è molto rigido e non evita la ridondanza dei dati.

**Modello reticolare:** è un’estensione del modello gerarchico. Questo modello consiste in due insiemi di dati, un insieme di record e un insieme di legami. Questo tipo di modello ha prestazioni più elevate perché è più vicino alla struttura fisica delle memorie di massa. Questo sistema si era diffuso nei sistemi di grandi dimensione perché il ritrovamento dei dati è veloce.

**Modello relazionale:** rappresenta il database come un insieme di tabelle. È il modello più vicino a modo di pensare i dati e si adatta alla classificazione e alla strutturazione dei dati.

## Termini specifici:

**Grado:** numero delle colonne della tabella.

**Dominio:** insieme dei valori che possono essere assunti da un attributo.

**Relazione:** insieme di n-uple, rappresentate con una tabella.

**Tupla:** riga di una tabella.

**Cardinalità:** numero delle n-uple.

**Chiave:** attributo che identifica univocamente le n-uple all’interno della relazione (ogni riga possiede valori diversi per l’attributo chiave).

## Requisiti che caratterizzano il modello relazionale:

1. Tutte le righe della tabella contengono lo stesso numero di colonne.
2. Gli attributi rappresentano informazioni elementari non scomponibili.
3. I valori assunti da un campo appartengono al dominio dei valori possibili per quel campo, e quindi sono omogenei tra di loro.
4. In una relazione ogni riga è diversa dalle altre, non ci possono essere 2 righe con gli stessi valori; quindi c’è un attributo che identificano la n-upla e quindi diventa chiave primaria.
5. Le n-uple compaiono secondo un ordine non prefissato.

**Integrità sull’entità:** nessuna componente della chiave primaria può avere valore nullo.

**Operazioni relazionali:** agiscono su una relazione per ottenere una nuova relazione, e cioè consentono di interrogare il database per avere le informazioni.

1. **Selezione:** genera una nuova relazione selezionando alcune tuple che soddisfano una condizione prefissata. Il grado è uguale a quello della relazione di partenza, e la cardinalità è uguale o minore.
2. **Proiezione:** genera una nuova relazione estraendo dalla tabella iniziale due o più colonne corrispondenti agli attributi prefissati. La tabella ottenuta potrebbe avere righe uguali ma devono essere eliminate. Il grado è minore o uguale a quello di partenza, la cardinalità è uguale a quella di partenza o minore quando ci sono righe uguali.
3. **Congiunzione:** serve a combinare due relazioni aventi uno o più attributi in comune, generando una nuova relazione che contiene le righe della prima e della seconda. Grado: se i gradi della relazione di partenza sono rispettivamente N1 e N2, il grado della nuova relazione è uguale a N1+N2-1; la cardinalità non è prevedibile.

## REGOLE DI DERIVAZIONE:

1. Ogni entità diventa una relazione.
2. Ogni attributo diventa un attributo della relazione, cioè il nome di una colonna.
3. Ogni attributo della relazione eredita le caratteristiche dell’attributo da cui deriva.
4. L’identificatore univoco diventa la chiave primaria della relazione derivata.
5. L’associazione 1:1 diventa un’unica relazione che contiene gli attributi della prima e della seconda entità.
6. L’identificatore univoco dell’associazione 1:N diventa chiave esterna dell’entità d arrivo associata.
7. L’associazione N:N diventa una nuova relazione composta dagli identificatori univoci delle due entità e dagli eventuali attributi dell’associazione.

## I linguaggi per i database

Le prestazioni del DBMS vengono attivare dall’utente usando appositi comandi, che costituiscono a tutti gli effetti un linguaggio, attraverso il quale l’utente può comunicare con il sistema di elaborazione che gestisce il database.

Tali comandi possono essere classificati nelle seguenti categorie:

* **DDL** (DataDefinitioLanguage): linguaggio per la descrizione dei dati, delle tabelle e delle viste, delle associazioni tra tabelle, dei vincoli di integrità e dei controlli relativi alla sicurezza.
* **DML** (Data Manipulation Language): linguaggio per il trattamento o manipolazione dei dati contenuti nel database, permette, cioè, le operazioni di accesso per inserimenti, modifiche o cancellazioni.
* **Query Language**: linguaggio per le interrogazioni alla base di dati.

La diffusione del modello relazionale ha favorito l’uso di linguaggi non procedurali, ma dichiarativi, formati da comandi specifici, per consentire agli utenti un facile uso delle prestazioni del DBMS. Tali linguaggi sono noti come Linguaggio per basi di dati ed hanno caratteristica di essere costituiti da un insieme completo di comandi che consente e facilita le operazioni di definizione dei database, di manipolazione dei dati, e di interrogazione da parte degli utenti. Nel linguaggio per basi di dati vengono, quindi, unificate in un unico linguaggio le funzioni dei linguaggi DDL, DML e Query Language.

Esempi di linguaggi per basi di dati sono:

* Il linguaggio SQL (Structured Query Language)
* QBE (Qyery By Example)

Il linguaggio SQL è divenuto ormai da tempo un linguaggio standard per creare, manipolare e interrogare database relazionali.

Il linguaggio SQL assolve le funzioni di:

* DDL che prevede istruzioni per definire la struttura delle relazioni delle basi di dati;
* DML che prevede istruzioni per manipolare i dati contenuti nelle diverse tabelle;
* Query Language che prevede istruzioni per poter interrogare la base di dati;

SQL può essere usato in:

* Modalità stand-alone o a sé stante;
* Modalità embedded o a linguaggio ospite;

## Sistemi informativi

Ogni organizzazione è dotata di un sistema informativo, che organizza e gestisce le informazioni necessarie per perseguire gli scopi dell’organizzazione stessa.

Funzioni di un sistema informativo:

* Raccolta, acquisizione delle informazioni;
* Archiviazione delle informazioni;
* Distribuzione, scambio di informazioni.

## Sistema Informatico

Il sistema informatico è la parte del sistema informativo che gestisce informazioni per mezzo della tecnologia informatica. Nei sistemi informatici, le informazioni vengono rappresentate in modo essenziale attraverso i dati.

Come per molti termini fondamentali, è difficile dare una definizione precisa del concetto di dato e informazione soprattutto delle differenze fra loro.

Definizioni:

* Informazione: Un dato che viene elaborato;
* Dato: elemento fondamentale dell’informatica;
* Archivio di dati: insieme organizzato di dati

## Il Progetto - La metodologia

Adottare una metodologia significa organizzare il lavoro. La metodologia è un approccio strutturato per l’esecuzione dell’intera attività di sviluppo dei progetto e dei relativi sistemi, coerentemente con gli obiettivi e le necessita da soddisfare.

L’organizzazione del lavoro di sviluppo di un progetto informatico assicura, oltre a un elevato gradi di qualità, anche un risparmio di risorse.

Le principali risorse, che beneficiano per il fatto di operare in un ambiente organizzato, cioè, definito, pianificato e coordinato, sono:

* Il tempo
* Le persone

E evidente che serve un minor tempo e un minor numero di persone se ognuno sa esattamente quando e cosa deve fare. Garantire la riuscita di un progetto informatico dipende quindi molto dalla struttura e della sequenza dei processi previsti per lo sviluppo, oltre che dall’impegno e dalle capacità delle persone che vi operano.

La metodologia indica al progettista il metodo da seguire per sviluppare il progetto. La costruzione di un progetto informatico si ottiene tramite l’esecuzione di diverse attività, caratterizzare da obiettivi ben precisi da raggiungere. La metodologia identifica queste attività e le organizza per ottimizzare il lavoro. Suddividiamo quindi la metodologia in fasi:

1. **Conoscenza degli obiettivi;**
2. **Analisi;**
3. **Progettazione/Transizione;**
4. **Realizzazione;**
5. **Documentazione/testing/formazione;**
6. **Esercizio.**

## Conoscenza degli obiettivi

Detta analisi dei requisiti, riguarda la conoscenza degli obiettivi. Contemporaneamente alla conoscenza degli obiettivi da raggiungere (il sistema da costruire) occorre la conoscenza globale dell’ambito di progetto (sistema esistente). Ciò si ottiene tramite l’indagine e l’approfondimento della situazione esistente. La rilevazione della situazione esistente si preoccupa di individuare in quale realtà di sistema informativo si deve integrare il nuovo progetto. Questa fase è anche chiamata analisi preliminare.

L’analisi preliminare deve anche tener conto di altri importanti aspetti riguardante l’uso del software:

* Sicurezza per la protezione dei dati da accessi non autorizzati;
* Riservatezza per la protezione dei dati sensibili;
* Ergonomia e usabilità del software per favorire il lavoro degli utenti finali.

Risultato di questa fase è:

* Documentazione del progetto ottenuta dalla ricerca e dallo studio della materia oggetto del progetto;
* Ipotesi di prodotto che consentirà di concretizzare l’idea del prodotto.

Per ottenere la conoscenza della materia oggetto del progetto è necessario poter disporre di esperti della materia ai quali fare l’intervista conoscitiva. Gli strumenti della tecnica dell’intervista sono:

* Le fonti della conoscenza (testi, persone esperte, coloro che svolgono mansioni inerenti l’area del progetto);
* Questionari (domande le cui risposte devono fornite tutte le informazioni necessari per documentare gli aspetti importanti);

## L’analisi

Ha l’obiettivo di determinare e di descrivere tutte le componenti del progetto. Questa fase del processo di sviluppo tratta la definizione di tre aspetti fondamentali:

* I dati, cioè le informazioni che caratterizzano il progetto;
* Funzioni, cioè le funzionalità richieste dal progetto;
* Flusso dei dati, cioè la modalità di acquisizione d i uscita dei dati (input/output rispetto alle funzioni.

## Progettazione

E la fase dove realmente si progetta il sistema. L’obiettivo della fase di progettazione è definire e descrivere dettagliatamente le caratteristiche del sistema informatico del progetto. I risultati principali di questa fase sono:

* Le specifiche per la creazione degli archivi elettronici;
* Le specifiche per lo sviluppo di moduli software del progetto.

## Transizione

L’obiettivo di questa fase è quello di definire la modalità di passaggio del sistema esistente al sistema del nuovo progetto.

## Realizzazione

Questa fase genera il prodotto del progetto.

## Documentazione

L’obiettivo di questa fase è fornire a tutti gli utenti del nuovo sistema il materiale documentato da consultare nel corso della loro attività lavorativa. Il prodotto della documentazione è composta dal manuale per l’utente.

## Testing

La finalità di questa fase è quella di verificare il grado di aderenza del progetto alle necessità indicate dall’utente-committente e di rimuovere eventuali errori presenti.

## L’esercizio o produzione

E la fase in cui si ha il rilascio del progetto. L’obiettivo di questa fase è attivare il progetto presso gli utenti, controllare che i processi elaborativi producano i risultati attesi e verificare globalmente il sistema.

## **La metodologia di progettazione**

Una metodologia di progettazione consiste in:

* Una descrizione dell’intera attività di progetto in passi successivi ed indipendenti tra loro;
* Una serie di strategie da seguire nei vari processi e alcuni criteri per la scelta in caso di alternative;
* Alcuni modelli di riferimento per descrivere i dati di ingresso e di uscita delle varie fasi.

Quindi, modellare i dati significa costruire una rappresentazione semplificata della realità osservata o di un problema aziendale, individuandone gli elementi caratterizzanti e i legami interconnessi tra loro.

La progettazione di un modello è articolata in tre fasi principali da effettuare a cascata:

1. **Progettazione concettuali:** rappresenta la realtà dei dati e le relazioni tra essi- Il prodotto di questa fase viene chiamato schema concettuale e fa riferimento a un modello concettuale dei dati.
2. **Progettazione logica:** consiste nella traduzione dello schema concettuale, definito nella fase precedente. Il prodotto di questa fase viene denominato schema logico della base di dati e fa riferimento a un modello logico dei dati secondo una rappresentazione ancora indipendente da dettagli fisici, ma concreta perché disponibile nel sistemi di gestione di base di dati.
3. **Progettazione fisica:** consiste nella rappresentazione dello schema logico per mezzo di strutture fisiche di memorizzazione. Il prodotto di questa fase viene chiamato anche fisico e fa riferimento a un modello fisico dei dati.

## Il modello concettuale

Il modello concettuale permetti di rappresentare i dati in modo indipendente da ogni sistema, cercando di descrivere i concetti del mondo reale.

Il più noto è il modello Entity-Relationship (E/R).

Esso è stato introdotto nel 1976 dal matematico Peter P. Chen, è uno strumento per analizzare le caratteristiche di una realità in modo indipendente dagli eventi che in essa accadono, cioè per costruire un modello concettuale dei dati indipendente dalle applicazioni.

Il risultato di questo lavoro è definizione di una rappresentazione grafica detta schema E/R.

La rappresentazione grafica è basata sull’Unified Modelling Language (UML). L’UML è un linguaggio grafico per visualizzare, specificare, costruire e documentare tutte le costruzioni di sistemi software.

## Costrutti del modello E/R

Gli elementi del modello E/R sono:

* Entità;
* Associazioni;
* Attributi.

**Entità** rappresentano classi di oggetti (fatti, cose, persone) che hanno proprietà comuni ed esistenza “autonoma” ai fini dell’applicazione di interesse. Quindi la caratteristica, dell’entità è che rappresenta oggetti con proprietà omogenee.   
Una **occorrenza o istanza** di un entità è un oggetto della classe che l’entità rappresenta.

Nella rappresentazione grafica le entità sono identificate con un rettangolo contenente all’interno il nome dell’entità.

**Associazioni (Relationship)** rappresentano legami logici, significativi per l’applicazione di interesse, tra due o più entità.

La rappresentazione grafica convenzionalmente usata per indicare un’associazione è un rombo che unisce le due entità interessate, il nome dell’associazione compare all’interno del rombo.

Nel linguaggio UML le relazioni tra tre o più entità sono rappresentate con un rombo al centro nel quale è riportato il nome dell’associazione e da linee che connettono il rombo con le entità-

**Attributi** descrivono la proprietà elementari di entità o associazioni che sono di interesse ai fini dell’applicazione.

Le caratteristiche di ogni attributo sono, **il formato, la dimensione e l’opzionalità**.

* Il formato di un attributo indica il tipo di valori che assume; i tre formati di base sono: carattere, numerico data/ore.
* La dimensione indica la quantità massima di caratteri o cifre inseribili;
* L’opzionalità indica la possibilità di non essere sempre valorizzato: l’attributo è obbligatorio se deve avere valore non nullo, facoltativo se sono accettabili valori nulli.

Graficamente gli attributi vengono elencati nella parte inferiore del rettangolo che rappresenta l’entità oppure collegata al rettangolo con linee e pallini.

Si indica con il termine **chiave o chiave primaria** un insieme minimale di attributi che permettono di distinguere tra loro le istanze di una stessa entità.

La chiave primaria di un’entità viene riconosciuta dalla presenza dell’acronimo {PK} posto tra parentesi graffe, accanto all’attributo chiave.

Nel caso di chiave formata da più attributi l’acronimo {PPK} è posto accanto a ognuno degli attributi che compongono la chiave.

## Molteplicità e Cardinalità delle associazioni

La **molteplicità** di un’associazione è il numero di possibili istanze che viene messo in corrispondenza con un’istanza dell’altra entità che partecipa all’associazione.

* Il numero minimo e il numero massimo di possibili istanze viene rappresentato mediate una coppia di valori separati da punti. ES. 1..1 , 0..1 , 1..N.
* Il valore minimo assume uno dei valori 0 e 1. Lo 0 indica che la partecipazione è facoltativa, mentre il valore 1 indica che la partecipazione è obbligatoria.
* Il valore massimo definisce la **cardinalità** della partecipazione nell’associazione. Esso assume uno dei due valori 1 o N per indicare una o molte partecipazioni all’associazione.

La cardinalità può quindi essere a uno oppure molti e pertanto le associazioni tra due entità si classificano nei seguenti tipi:

1. Associazione **uno a uno** o biunivoca, indicata con **1:1**;
2. Associazione **uno a molti,** indicata con **1:N**;
3. Associazione **molti a molti**, indicata con **N:N**;

## Relazioni e tabelle

In matematica dati due insiemi D1 e D2, si chiama prodotto cartesiano di D1 e D2, in simboli D1xD2, l’insieme delle coppie ordinate (N1,N2), tali che N1 è un elemento di D1 e N2 è un elemento di D2.

ES. A {1,2,3,4} e B{a,b}

AxB = {(1,a),(1,b), (2,a),(2,b), (3,a),(3,b), (4,a),(4,b)}

La **relazione matematica** sugli insieme D1 e D2 (chiamati domini della relazione) è un sottoinsieme di D1xD2.

Il numero n delle componenti del prodotto cartesiano (domini) viene detto **grado** del prodotto cartesiano e della relazione. Il numero di elementi (cioè di n-ple) della relazione viene chiamato **cardinalità** della relazione.

Una relazione è un insieme, quindi:

* Non è definito alcun ordinamento fra le n-ple;
* Le n-ple di una relazione sono distinte l’una dall’altra.

## Algebra relazionale

Insieme di operatori:

* Su relazioni;
* Che producono relazioni;
* E quindi possono essere la base per espressioni complesse.

Operatori dell’algebra relazionale:

* **Unione, intersezione, differenza;**
* **Ridenominazione;**
* **Selezione;**
* **Proiezione;**
* **Join (join naturale, prodotto cartesiano, theta-join)**

Operatori insiemistici:

* Le relazioni sono insiemi;
* I risultati debbono essere relazioni (insieme di ennuple omogenee)

Quindi;

* È possibile applicare gli operatori insiemistici (unione, intersezione, differenza) solo a coppie di relazioni definite sugli stessi attributi.

**Ridenominazione**

E’ un operatore monadico (unitario). Intuitivamente, modifica lo schema lasciando inalterata l’istanza dell’operando. Permette di superare le limitazioni imposte agli operatori insiemistici.

**Selezione**

E’ un operatore monadico che produce un risultato che:

* Ha lo stesso schema dell’operando;
* Contiene un sottoinsieme delle ennuple dell’operando;

Produce decomposizioni orizzontali.

Il risultato di una selezione contiene le ennuple dell’operando che soddisfano la condizione di selezione.

**Proiezione**

E’ un operatore monadico.

Produce un risultato:

* Definito su un sottoschema dell’operando;
* A cui contribuiscono tutte le ennuple dell’operando.

Produce decomposizioni verticali.

Il risultato di una proiezione contiene al più tante ennuple quante l’operando;

Ma può contenerne di meno.

**Join naturale**

E’ un operatore binario (generalizzabile).

Correla dati di relazioni diverse.

E’ l’operatore più caratteristico dell’algebra relazionale.

Programmazione (SQL, PHP, HTML)

## SQL attributi tipo standard:

* **BOOLEAN** – Valore logico (true, false);
* **CHARACTER(n**) – Stringa di lunghezza n (n da 1 a 15000);
* **DATE** – Data nella forma MM/GG/AA;
* **TIME** – Ora nella forma HH:MM;
* **INTEGER(p)** – Numero intero con precisione 10 (da -32768 a 32767)
* **DECIMAL(p,s)** – Numero decimale con precisione p e s sono cifre decimali;
* **REAL** – Numero reale con mantissa di precisione 7;
* **FLOAT** – Numero reale con mantissa di precisione 15;
* **FLOAT(p)** – Numero reale con mantissa di precisione p.

## Esempio creazione tabella

Esempio:

CREATE TABLE DIPARTIMENTO(

Codice char(5),

Descrizione char(29) not null,

Sede char(20),

Manager smallint,

Primary key (codice),

Unique(descrizione),

Foreign Key (Manager) references IMPIEGATO(ID)

On Delete se null,

On Update cascade

);

## Esempio Pagina PHP

<?php

$matricola= $\_GET["matricola"];

$nome = $\_GET["nome"];

$cognome = $\_GET["cognome"];

$indirizzo = $\_GET["indirizzo"];

$voto = $\_GET["voto"];

**$conn = mysqli\_connect("Host","nome\_account","passoword","Nome\_Databse");**

if(!$conn){

echo "Errore durante la connessione <br>";

exit();

}

echo "SONO CONNESSO ALL'ACCOUNT! <br>";

echo " SONO CONNESSO AL DATABASE! <br>";

echo "HAI APPENA AGGIUNTO I SEGUENTI DATI:<br>";

echo "<br>$matricola<br> $nome<br> $cognome<br> $indirizzo<br> $voto <br>";

$query= "INSERT INTO STUDENTE(matricola, nome, cognome, indirizzo, voto) VALUES ('$matricola','$nome','$cognome','$indirizzo','$voto');";

if(!**mysqli\_query($conn,$query**)){

echo "Errore comando insert <br>";

exit();

}

echo "Studente aggiunto correttamente <br>";

$query = "Select \* from STUDENTE;";

**$ris=mysqli\_query($conn,$query);**

**$riga = mysqli\_fetch\_array($ris);**

if(!$riga){

echo "errore comando fetch";

exit();

}

echo "<h3>DATI DELLE PERSONE SCHEDATE!</h3><br>";

echo " <table> <tr> <td> matricola </td><td> nome </td><td> Cognome </td><td> Indrizzo </td><td> voto </td> </tr> ";

while($riga){

echo "<tr><td>".$riga['matricola']."</td>";

echo "<td>".$riga['nome']."</td>";

echo "<td>".$riga['cognome']."</td>";

echo "<td>".$riga['indirizzo']."</td>";

echo "<td>".$riga['voto']."</td></tr>";

**$riga = mysqli\_fetch\_array($ris);**

}

echo "</table>";

**mysql\_close($conn);**

?>

## Esempio pagina WEB dinamica

<html>

<head>

<title>CENTRO SPIONAGGIO CINESE</title>

</head>

<body>

<h1> TUTTI I TUOI DATI SARANNO SPEDITI AL PARTITO</h1>

<**form** **method**="GET" **action**="main.php">

matricola <input type=text name=matricola> <br>

Nome <input type=text name=nome> <br>

Cognome <input type=text name=cognome> <br>

Indirizzo <input type=text name=indirizzo> <br>

voto <input type=text name=voto> <br>

<input type=submit value="SPEDISCI">

</form>

</body>

</html>