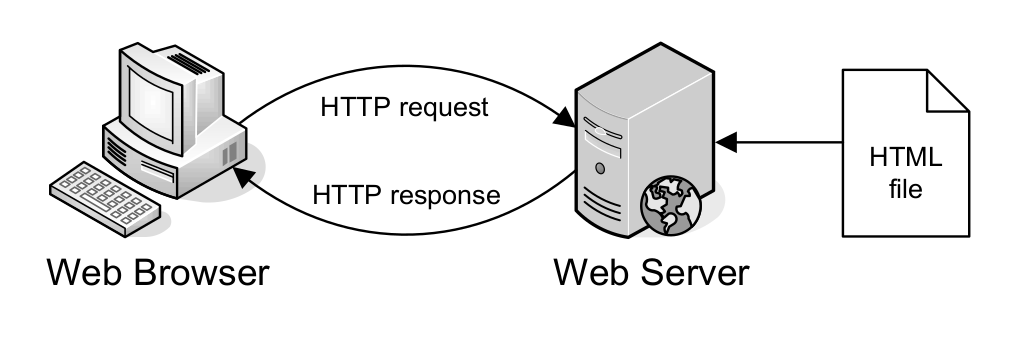
**Lezione 14/03/2024**

da: <http://www.mauro.bogliaccino.it/public/fondamenti/md/Fondamenti-master-019_static_dynamic.md>

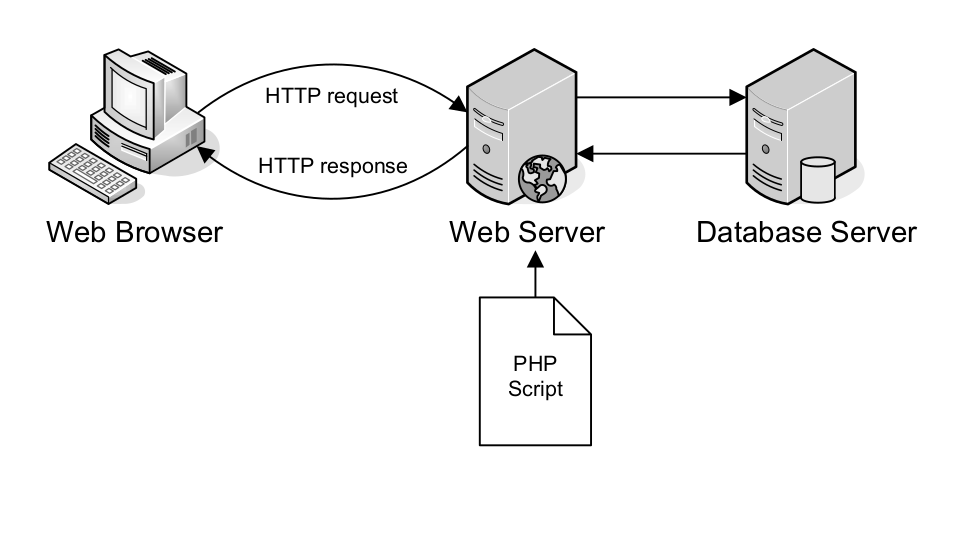
**Pagine web statiche:**

****

**‧**se la metto sul server(computer proiettato su internet-web server- può essere raggiunta)

**‧**se c’è lo dà altrimenti errore 404

**Pagine web dinamiche:**



**‧**comunicazione client-server medesimo a quello statico

**‧**il server è dotato di una programmazione (php, asp,...) che gli consente di essere affiancato da un server database dove risiede i dati è sono più facili da reperire e visualizzare (fa richiesta, letta da php…, chiama database, database da dati, web server risponde con pagina html+tabella dati: tutto dinamico, tutto codificato dai server)

| **PDV Rete locale:** | **PDV + ampio (Internet):** |
| --- | --- |
|  |  |

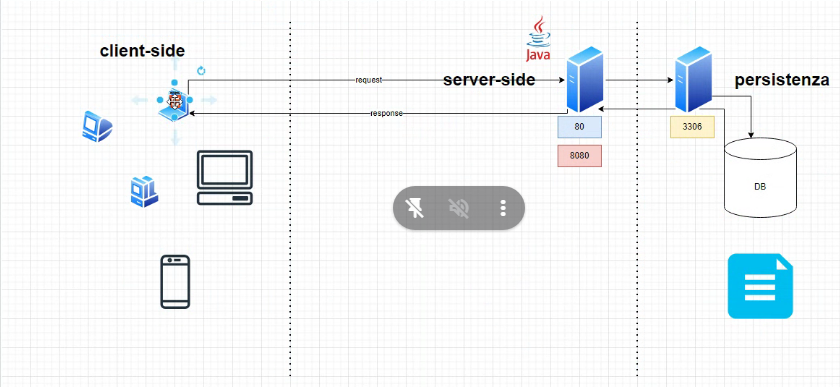
cosa c'entra programmazione web con sql, ecc?

dati inseriti nei database(ricchezza),

l’accesso è controllato con logica server-side, presenza di un server con servizio specifico (accesso con porta specifica es port:80, 3306….) cmd: netstat -an (uso di java,long e tonkat)

successivamente ci sono i client(pc, smartphone…. ) che fanno richieste ai server con un indirizzo,

poi server db check sicurezza e poi invia al web server che prepara bene html e invia risposta (da web-side a client-side)



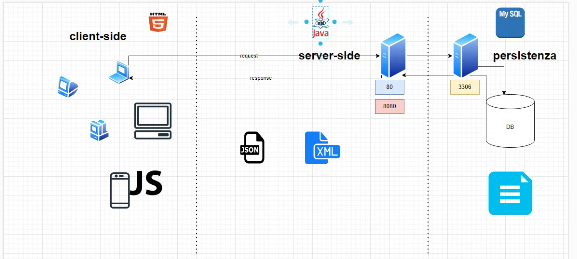
il problema sono le dimensioni schermo del device,

motivo: costruzione sito da server-side

un altro problema è anche il tempo di elaborazione del file che non consente la visualizzazione rapidissima da molti device

soluzione scinde pagina fatta dal server scissa e invia solo i dati in un formato(json, xml) che attraverso un javascript viene decodificato e la pagina web viene prodotta lato client

nb rest: uso di un formato intermediario per comunicare dati tra client e server(json x es) e viene letto da JS



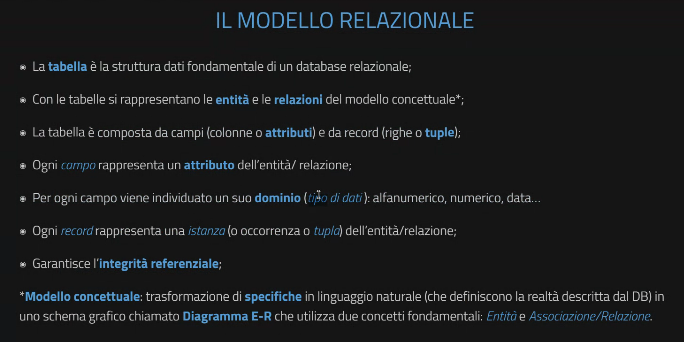
**COMANDI:**

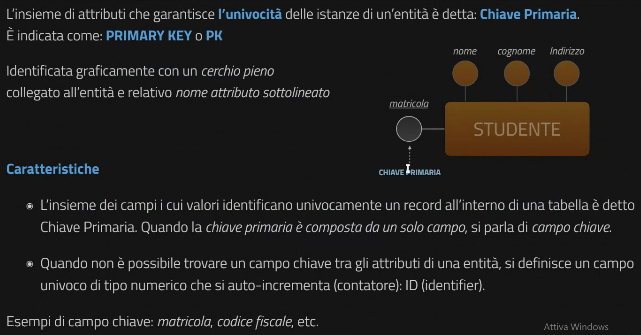
[**https://github.com/maboglia/DataBase/blob/main/02\_SQL\_Fondamenti/01\_DDL\_DML.md**](https://github.com/maboglia/DataBase/blob/main/02_SQL_Fondamenti/01_DDL_DML.md)

**ALGEBRA RELAZIONALE >> MODELLO RELAZIONALE:**

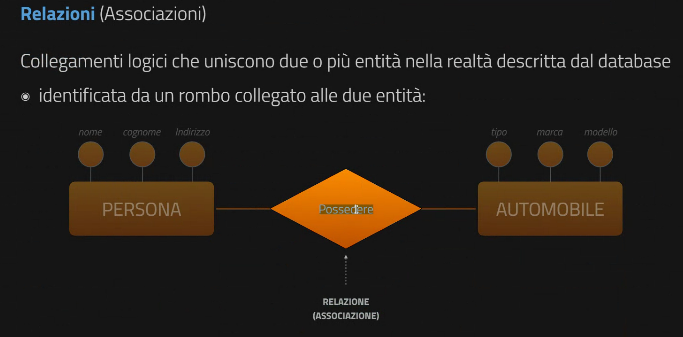
[**https://github.com/maboglia/DataBase/blob/main/01\_AlgebraRelazionale/01\_alegbra\_relazionale.md**](https://github.com/maboglia/DataBase/blob/main/01_AlgebraRelazionale/01_alegbra_relazionale.md)

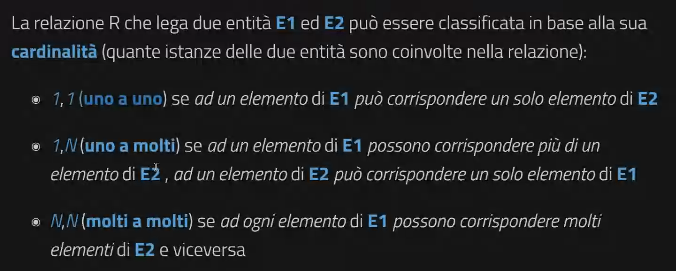
[**https://github.com/maboglia/DataBase/blob/main/01\_AlgebraRelazionale/02\_modello\_relazionale.mdù**](https://github.com/maboglia/DataBase/blob/main/01_AlgebraRelazionale/02_modello_relazionale.md%C3%B9)





Entità = tabelle con attributi, quelli che garantiscono univocità sono chiavi primarie

****

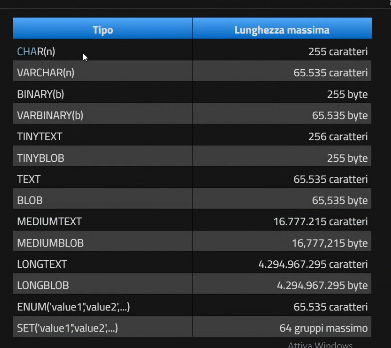
****

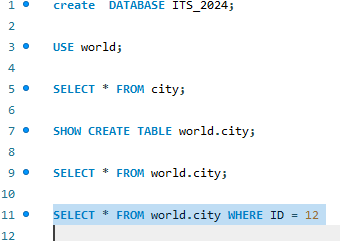
**da chatgpt:**

* First Normal Form (1NF):
  + Questa forma richiede che ogni attributo di una tabella contenga solo valori atomici, cioè valori che non possono essere suddivisi ulteriormente. In altre parole, non dovrebbero esserci array o strutture complesse all'interno di un attributo.
  + Ogni colonna in una tabella deve contenere valori atomici e ogni cella deve contenere un solo valore.
* Second Normal Form (2NF):
  + Per soddisfare la 2NF, la tabella deve già essere in 1NF. Inoltre, ogni attributo non chiave deve dipendere completamente dalla chiave primaria.
  + Se una tabella ha una chiave composta, ogni attributo non chiave deve dipendere da tutte le parti di quella chiave, non solo da una parte di essa.
* Third Normal Form (3NF):
  + Per soddisfare la 3NF, la tabella deve già essere in 2NF. Inoltre, ogni attributo non chiave deve dipendere solo dalla chiave primaria e non da altri attributi non chiave.
  + Se un attributo non chiave dipende da un altro attributo non chiave, dovrebbe essere separato in una tabella separata e relazionato con una chiave esterna.

In sintesi, mentre la 1NF si concentra sulla struttura atomica dei dati, la 2NF e la 3NF si concentrano sulla ridondanza e sulla dipendenza dei dati. Normalizzando il database in queste forme, si migliora la struttura del database e si riducono i problemi di aggiornamento, cancellazione e inserimento anomalo dei dati, garantendo una migliore manutenzione e scalabilità del sistema.

NB se trovi il ppt è meglio





CREATE TABLE `its\_2024`.`corsi` (

`id` INT NOT NULL AUTO\_INCREMENT,

`nome` VARCHAR(45) NULL,

`ore` TINYINT(4) UNSIGNED NULL,

PRIMARY KEY (`id`));