**Lezione 21/03/2024**

**elementi di foglio elettronico**

Uso di tabelle per caricamento di dati in database

permette di fare varie cose: calcolo usando dati, visualizzazione dati….

nb: nelle tabelle non tutti gli oggetti sono entità (da tab registro prof), possono essere anche relazioni(?)

in un mondo piatto (flat) come i fogli di calcolo ecc, è indispensabile usare un modello **entità-relazioni** per gestire meglio i dati di un file

**reference:** riferimento, collegamento ad un oggetto nella tabella di dati,

(es un libro segnato in uno schedario)

un oggetto può avere più reference (1 libro presente in schedarioA(legge autori) e schedarioB(legge data))

<https://github.com/maboglia/DataBase/blob/main/02_SQL_Fondamenti/00_fondamenti-DB2.md>(tutto)

**Tabelle e Record**

Parliamo ora di "Tabelle e Record". Le tabelle costituiscono uno degli elementi chiave in un database, fornendo una struttura organizzata per immagazzinare i dati.

Ogni tabella è composta da righe e colonne, dove ogni riga rappresenta un record e ogni colonna rappresenta un attributo o una caratteristica specifica dei dati.**Record**

Record: Un record è un'istanza di dati rappresentata da una singola riga all'interno di una tabella. Ad esempio, in una tabella che gestisce informazioni sugli studenti, ogni record potrebbe contenere i dettagli specifici di uno studente, come il nome, il cognome, il numero di matricola e altre informazioni.**Campi o Colonne**

Campi o Colonne: Le colonne di una tabella rappresentano gli attributi o le proprietà dei dati. Ad esempio, in una tabella degli studenti, potrebbero esserci colonne come "Nome", "Cognome", "Numero di Matricola", "Corso di Studi", ecc. Ogni colonna contiene dati di un tipo specifico, come stringhe di testo, numeri o date.

**Chiavi**

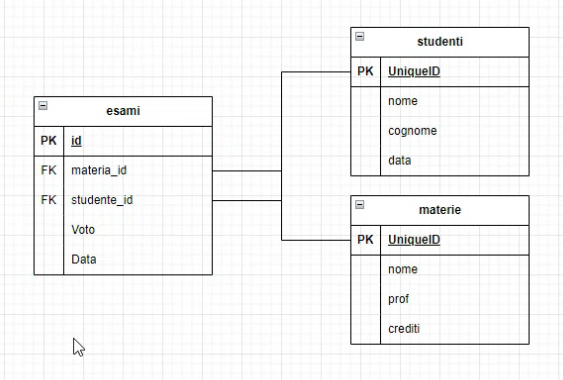
Chiavi: Le chiavi sono elementi cruciali nella progettazione di una tabella. La chiave primaria identifica in modo univoco ogni record nella tabella. Ad esempio, un numero di matricola potrebbe essere utilizzato come chiave primaria in una tabella degli studenti. Le chiavi esterne sono usate per stabilire relazioni tra tabelle.

**Relazioni tra Tabelle**

Relazioni tra Tabelle: Nei database relazionali, le tabelle sono spesso collegate attraverso relazioni. Ad esempio, una tabella degli studenti potrebbe essere collegata a una tabella dei corsi tramite una chiave esterna, indicando quali studenti sono iscritti a quali corsi.**Attributi e Tipi**

Attributi e Tipi di Dati: Ogni colonna di una tabella ha un tipo di dato associato, come VARCHAR per stringhe di testo, INTEGER per numeri interi, DATE per date, ecc. La definizione accurata degli attributi e dei tipi di dati è essenziale per garantire la coerenza dei dati.La progettazione delle tabelle è un aspetto critico nella creazione di un database efficiente. Una buona progettazione delle tabelle contribuisce a garantire l'integrità dei dati, la facilità di interrogazione e una gestione efficiente dei dati all'interno del sistema di gestione di database**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)

**TIPI DI DATO**[**https://github.com/maboglia/DataBase/blob/main/02\_SQL\_Fondamenti/01\_tipi\_di\_dato.md**](https://github.com/maboglia/DataBase/blob/main/02_SQL_Fondamenti/01_tipi_di_dato.md)

**Dati da MySQL **

esami è una tabella relazionale

elaborato con metodo rdbms

create table esami (

id int not null auto\_increment,

studente\_id int not null,

corso\_id int not null,

voto int not null,

data\_esame date,

primary key (id)

);

drop table esami;

describe corsi;

desc esami;

desc studenti;

show create table corsi;

truncate table corsi;

select \* from corsi;

alter table corsi add prof varchar(50);

insert into corsi (nome, prof, ore) values

('programmazione java', 'mauro', 100),

('fondamenti basi di dati', 'mauro', 100),

('inglese', 'elisabetta', 100),

('fondamenti di programmazione', 'valentino', 100);

truncate corsi;

select nome from its\_2024.studenti order by nome;

select nome from its\_2024.studenti order by cognome;

select nome, cognome from its\_2024.studenti order by cognome;

select cognome, nome from its\_2024.studenti order by cognome;

select cognome, nome from its\_2024.studenti where id =1;

select \* from corsi where id =1;

select \* from esami;

select studenti.cognome, corsi.nome, esami.data\_esame, esami.voto

from esami, studenti, corsi

where studenti.id=esami.studente\_id

and corsi.id=esami.corso\_id;