VISTE

Viste

- Le Viste Logiche o Viste o View possono essere definite come delle tabelle virtuali, i cui dati sono riaggregazioni dei dati contenuti nelle tabelle “fisiche” presenti nel database.
- Le tabelle fisiche sono gli unici veri contenitori di dati.
- Le viste non contengono dati fisicamente, ma forniscono una diversa visione, dinamicamente aggiornata, dei dati presenti nelle tabelle dati.

- La vista appare all’utente come una normale tabella, in cui può effettuare interrogazioni e, limitatamente ai suoi privilegi, anche modifiche dei dati (ma non ne parleremo).

DDL: Viste (riassumendo)

- Viste
  - tabelle “virtuali”
  - definite attraverso un’interrogazione
  - possono essere utilizzate come tabelle reali

- Due funzioni fondamentali:
  - creazione degli schemi esterni (es. per la privatezza dei dati)
  - semplificazione di interrogazioni ricorrenti

Viste, sintassi

- Il comando DDL che consente di definire una vista ha la seguente sintassi

```
CREATE VIEW NomeVista [ ( ListaAttributi ) ] AS SelectSQL
[ with [ local | cascaded ] check option ]
```

- I nomi delle **colonne** indicati nella **lista attributi** sono i nomi assegnati alle **colonne della vista**, che corrispondono ordinatamente alle colonne elencate nella select.
- Se questi non sono specificati, le colonne della vista assumono gli stessi nomi di quelli della/e tabella/e a cui si riferisce.

DDL: Viste

- Creazione di viste
  - CREATE VIEW <nome> AS <SELECT>;
- Semantica
  - la vista viene ricalcolata sulla base della sua definizione ogni volta che viene usata
- Eliminazione di viste
  - DROP VIEW <nome>;

Create View, Esempio 1

- Creare una vista contenente il nome, l'età e lo stipendio degli impiegati del dipartimento di Produzione il cui stipendio è maggiore di 2500 euro

```
Create view ImpiegatiProduzione
      (Nome, Eta, Stipendio) AS
      Select Nome, Eta, Stipendio
      From Impiegato
      Where Dipart = Produzione'
            and Stipendio > 2500
```

Impiegato

NOME	ETA	STIPENDIO	DIPART
Andrea	27	2100	Direzione
Aldo	25	1500	Produzione
Maria	55	4200	Distribuzione
Anna	50	3500	Produzione
Filippo	26	3900	Distribuzione
Luigi	50	4000	Amministrazione
Franco	60	2000	Produzione
Olga	30	4100	Produzione
Sergio	85	3500	Amministrazione
Luisa	75	8700	Direzione

DDL: Viste – Esempio 2

- Privatezza: esami senza voti.
- Visualizzare i dati degli studenti con i relativi agli esami sostenuti senza voti

```
CREATE VIEW EsamiSenzaVoti AS
SELECT studente, corso
FROM Esami;

SELECT * FROM EsamiSenzaVoti;

SELECT *
FROM Studenti, EsamiSenzaVoti
WHERE matr=studente;

DROP VIEW EsamiSenzaVoti;
```

Esami

studente	corso	voto	lode
111	PR1	27	false
222	ASD	30	true
111	INFT	24	false
77777	PR1	21	false
77777	ASD	20	false
88888	ASD	28	false
88888	PR1	30	false
88888	INFT	30	true

Studenti

matr	cognome	nome	ciclo	relat
111	Rossi	Mario	laurea tr.	null
333	Rossi	Maria	laurea tr.	null
222	Neri	Paolo	laurea tr.	null
444	Pinco	Palla	laurea tr.	FT
77777	Bruno	Pasquale	laurea sp.	FT
88888	Pinco	Pietro	laurea sp.	CV

DDL: Viste – Esempio 3

Professori

cod	cognome	nome	qualifica	Dipartimento
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Informatica
ADP	Del Piero	Alessandro	supplente	null

Numeri

professore	numero
FT	0971205145
FT	347123456
VC	0971205227
ADP	0971205363
ADP	338123456

- Scrivere una vista che associ ad ogni professore i suoi numeri di telefono e ne visualizzi il contenuto. Visualizzare codice, nome, cognome, numero.

```
CREATE VIEW ProfessoriNumeri AS
    SELECT codice, nome, cognome, numero
    FROM Professori JOIN Numeri
        ON cod=professore;

SELECT *
FROM ProfessoriNumeri
ORDER BY cognome, nome;
```

codice	cognome	nome	numero
…	…	…	…
…	….	…	…

DDL: Viste

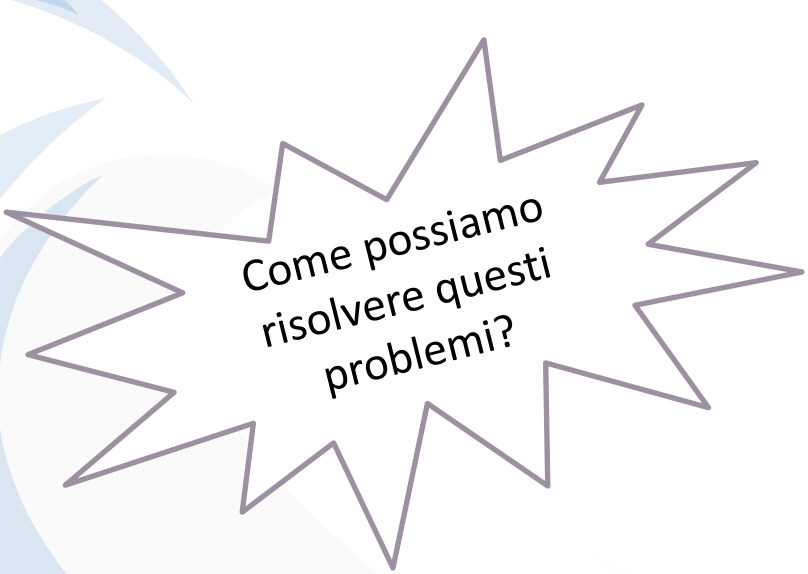
- Differenza tra tabelle e viste
  - le tabelle sono materializzate nella base di dati, le viste no (sono derivate dalle tabelle)
  - schema di una vista = attributi e tipi della select
  - istanza di una vista = risultato della select
  - le tabelle sono aggiornabili, le viste no
  - le viste sono sempre aggiornate e consistenti
  - non hanno impatto sulle prestazioni

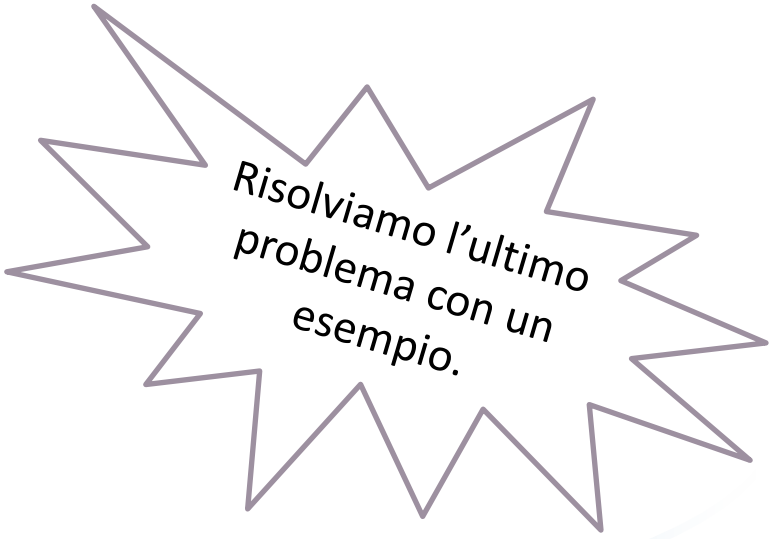
Potere espressivo in SQL

- Il linguaggio SQL non permette di scrivere espressioni equivalenti a tutte le funzioni calcolabili (non ha la potenza computazione delle Macchine di Turing).
- Esempi di interrogazione non esprimibili in SQL:
  - Le funzioni di aggregazione di solito disponibili non sono del tutto sufficienti, ad esempio non vi sono le funzioni di tipo statistico, come la varianza di una colonna di valori.
  - Non è possibile scrivere query che direttamente consentano di risolvere problemi di information retrieval. Esempio:
    - Documenti (codice, titolo, autore)
    - ParoleChiavi (codice, parolaChiave)
    - E' possibile trovare i titoli dei documenti che contengono una certa parola chiave oppure che contengono tutte le parole chiavi di un certo insieme K
    - Non è possibile scrivere un'interrogazione per trovare tutti i documenti che contengono almeno k delle chiavi di K.

Potere espressivo in SQL

- Altri esempi di interrogazione non esprimibili in SQL:
  - La chiusura transitiva di una relazione binaria rappresentata in forma di tabella non può essere calcolata.
  - Le funzioni di aggregazione non si possono applicare ad altre funzioni. Per esempio, non si può calcolare il totale di tutte le medie dell'ammontare degli ordini degli agenti di vendita, o il massimo di tutti i loro totali di vendita.





Nidificazione, Viste, Potere Espressivo

- Esempio:
  - Restituire il numero medio di docenti per dipartimento

Professori

cod	cognome	nome	qualifica	Dipartimento
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Informatica
ADP	Del Piero	Alessandro	supplente	null
…	…	…	…	…

~~SELECT avg(count(cod))~~
~~FROM Professori~~
~~GROUP BY dipartimento;~~

```
CREATE VIEW Dipartimento AS
SELECT dipartimento, count(*) AS numdocenti
FROM Professori
GROUP BY dipartimento;
```

```
SELECT avg(numdocenti)
FROM Dipartimento;
```

Nidificazione e Viste

- Esempio di nidificazione nelle viste

- Esempio: Studenti con la media più alta
  - per calcolare la media di ciascuno studente serve un raggruppamento
  - condizione nidificata sui gruppi
  - non è possibile nidificare la HAVING (nidificazione solo nella WHERE)

Nidificazione, Viste, Potere Espressivo

- Esempio:
- Studenti con la media più alta

Studenti

matr	cognome	nome	ciclo	anno	relatore
111	Rossi	Mario	laurea tr.	1	null
222	Neri	Paolo	laurea tr.	2	null
333	Rossi	Maria	laurea tr.	1	null
444	Pinco	Palla	laurea tr.	3	FT
77777	Bruno	Pasquale	laurea sp.	1	FT
88888	Pinco	Pietro	laurea sp.	1	CV

Esami

studente	corso	voto	lode
111	PR1	27	false
222	ASD	30	true
111	INFT	24	false
77777	PR1	21	false
77777	ASD	20	false
88888	ASD	28	false
88888	PR1	30	false
88888	INFT	30	true

~~max(avg(voto))~~ ????

Nidificazione, Viste, Potere Espressivo

- Esempio: Studenti con la media più alta
- Soluzione con le viste

```
CREATE VIEW StudentiConMedia AS
SELECT matr, cognome, nome, avg(voto) as media
FROM Esami JOIN Studenti ON studente=matr
GROUP BY matr, cognome, nome;
```

```
SELECT matr, cognome, nome
FROM StudentiConMedia
WHERE media = (SELECT max(media)
               FROM StudentiConMedia);
```

StudentiConMedia

matr	cognome	nome	media
111	Rossi	Mario	20,7
222	Neri	Paolo	24,5
333	Rossi	Maria	25,8
444	Pinco	Palla	19,6
77777	Bruno	Pasquale	26
88888	Pinco	Pietro	26

Le VIEW possono essere usate come tabelle

- Le VIEW possono essere distrutte alla pari di tabelle
  - DROP (TABLE | VIEW) Nome [RESTRICT|CASCADE]
    - Con RESTRICT non viene cancellata se è utilizzata in altre viste
    - Con CASCADE verranno rimosse tutte le viste che usano la View o la Tabella rimossa.

  - La distruzione di una VIEW non altera le tabelle su cui la VIEW si basa.

Domanda 11

- Squadre (CodiceSquadra, NomeSquadra, AnnoFondazione, Regione)
  Allenatori (Matricola*, codiceSquadra*)
  Persone (Matricola, Nome, Cognome, RedditoAnnuo, cittaNascita)
  Giocatori (Matricola*, Ruolo, codiceSquadra, NumeroReti)
  Partita (SquadraInCasa*, SquadraFuoriCasa*, data)

- Visualizzare il codice della squadra che ha realizzato il massimo numero di gol.

Domanda 12

- Squadre (CodiceSquadra, NomeSquadra, AnnoFondazione, Regione)
  Allenatori (Matricola*, codiceSquadra*)
  Persone (Matricola, Nome, Cognome, RedditoAnnuo, cittaNascita)
  Giocatori (Matricola*, Ruolo, codiceSquadra, NumeroReti)
  Partita (SquadraInCasa*, SquadraFuoriCasa*, data)

- Visualizzare il nome della squadra con la media PIU' ALTA dei redditi dei giocatori. Visualizzare anche la relativa media.