

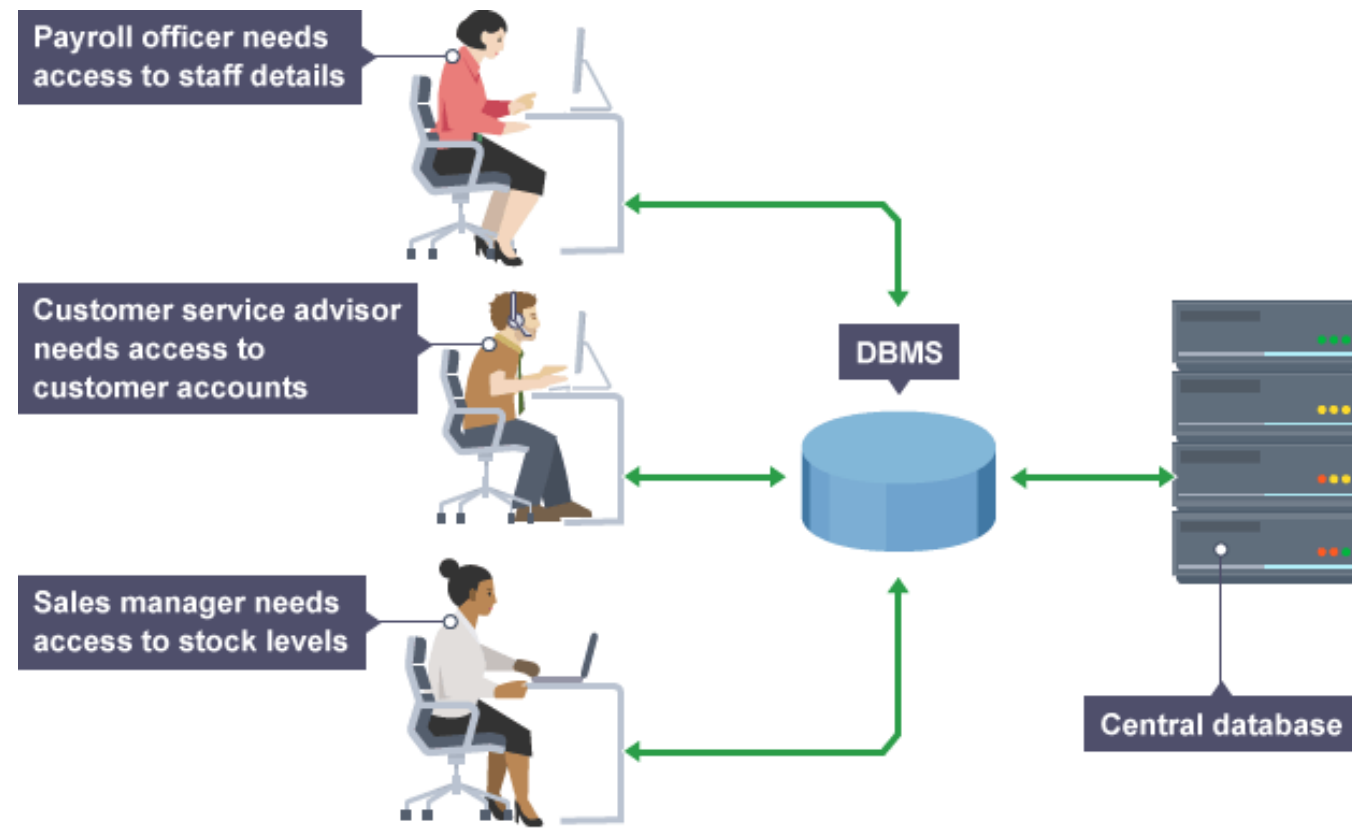
A decorative graphic on the left side of the slide, consisting of a network of blue and green lines and small circles, resembling a circuit board or a data network.

Introduzione di database e agli indici

In informatica, il termine **database**, indica un insieme organizzato di dati.

Un **DBMS** (Data Base Management System) è un sistema di gestione delle basi di dati che permette diverse operazioni:

- Creare archivi specificando dati e relative regole di utilizzo
- Inserire, cancellare e modificare dati
- Effettuare **Query** per ricercare ed estrarre dati d'interesse



Modello Relazionale RDBMS (Relational DBMS)

Il **modello relazionale**, come dice il termine stesso, si basa su **relazioni univoche** fra i dati. In pratica, tutti i dati da trattare sono memorizzati in **strutture fisse**, dette **tabelle**, dove sono posti l'uno di seguito all'altro a formare **tuple** (ossia le **righe** di ciascuna tabella). Ogni dato contenuto in una tabella si trasforma in un'informazione vera e propria solo se associato al suo corrispondente **attributo**, che rappresenta l'intestazione della **colonna** in cui il dato è memorizzato.

| Id | Nome | Cognome | Fascia | Id_afferenza | Orcid |
|----|------------|-----------|-------------|--------------|-------|
| 1 | Beatrice | ACCIAIO | Ricercatore | 620 | 0 |
| 2 | Alessandro | ACHILLI | Associato | 619 | 0 |
| 3 | Gabriele | ACUTI | Ricercatore | 630 | 0 |
| 4 | Alberto | AGNELLI | Associato | 631 | 0 |
| 5 | Giancarlo | AGNELLI | Ordinario | 628 | 0 |
| 6 | Emidio | ALBERTINI | Associato | 631 | 0 |

Query

- Il linguaggio di interrogazione usato nei database relazionali e' **SQL**.

```
SELECT [ ALL | DISTINCT ] lista_elementi_selezione
FROM lista_riferimenti_tabella
[ WHERE espressione_condizionale ]
[ GROUP BY lista_colonne [HAVING Condizione] ]
[ ORDER BY lista_colonne ];
```

| Id | Nome | Cognome | Fascia | Id_afferenza | Orcid |
|----|-------------------|----------------|-------------|--------------|-------|
| 1 | Beatrice | ACCIAIO | Ricercatore | 620 | 0 |
| 2 | <u>Alessandro</u> | <u>ACHILLI</u> | Associato | 619 | 0 |
| 3 | Gabriele | ACUTI | Ricercatore | 630 | 0 |
| 4 | <u>Alessandro</u> | <u>ACHILLI</u> | Associato | 631 | 0 |
| 5 | Giancarlo | AGNELLI | Ordinario | 628 | 0 |
| 6 | Alberto | AGNELLI | Ricercatore | 631 | 0 |

- Esempio di query SQL.

```
SELECT DISTINCT Nome, Cognome
FROM Autori
WHERE Id < 7
ORDER BY Nome
```

| Nome | Cognome |
|-------------------|----------------|
| Alberto | AGNELLI |
| <u>Alessandro</u> | <u>ACHILLI</u> |
| Beatrice | ACCIAIO |
| Gabriele | ACUTI |
| Giancarlo | AGNELLI |

Vincoli di referenzialità: Creazione chiavi

```
create table Autori (Id int,  
                    Nome varchar(1000),  
                    Cognome varchar(1000),  
                    Fascia varchar(1000),  
                    Id_afferenza int,  
                    Orcid varchar(1000),  
                    primary key (Id),  
                    foreign key (Id_afferenza) references Atenei_dipartimenti (Id),  
                    index index_cognome (Cognome));
```

► Autori

| Colonna | Tipo | Commento |
|---------------------|---------------|----------|
| Id | int(11) | |
| Nome | varchar(1000) | |
| Cognome | varchar(1000) | |
| Fascia | varchar(1000) | |
| Id_afferenza | int(11) | |
| Orcid | varchar(1000) | |

► Atenei_Dipartimenti

| Colonna | Tipo | Commento |
|---------------------|---------------|----------|
| Id | int(11) | |
| Ateneo | varchar(1000) | |
| Dipartimento | varchar(1000) | |
| Regione | varchar(1000) | |
| Nazione | varchar(1000) | |
| Ring_gold_id | varchar(1000) | |

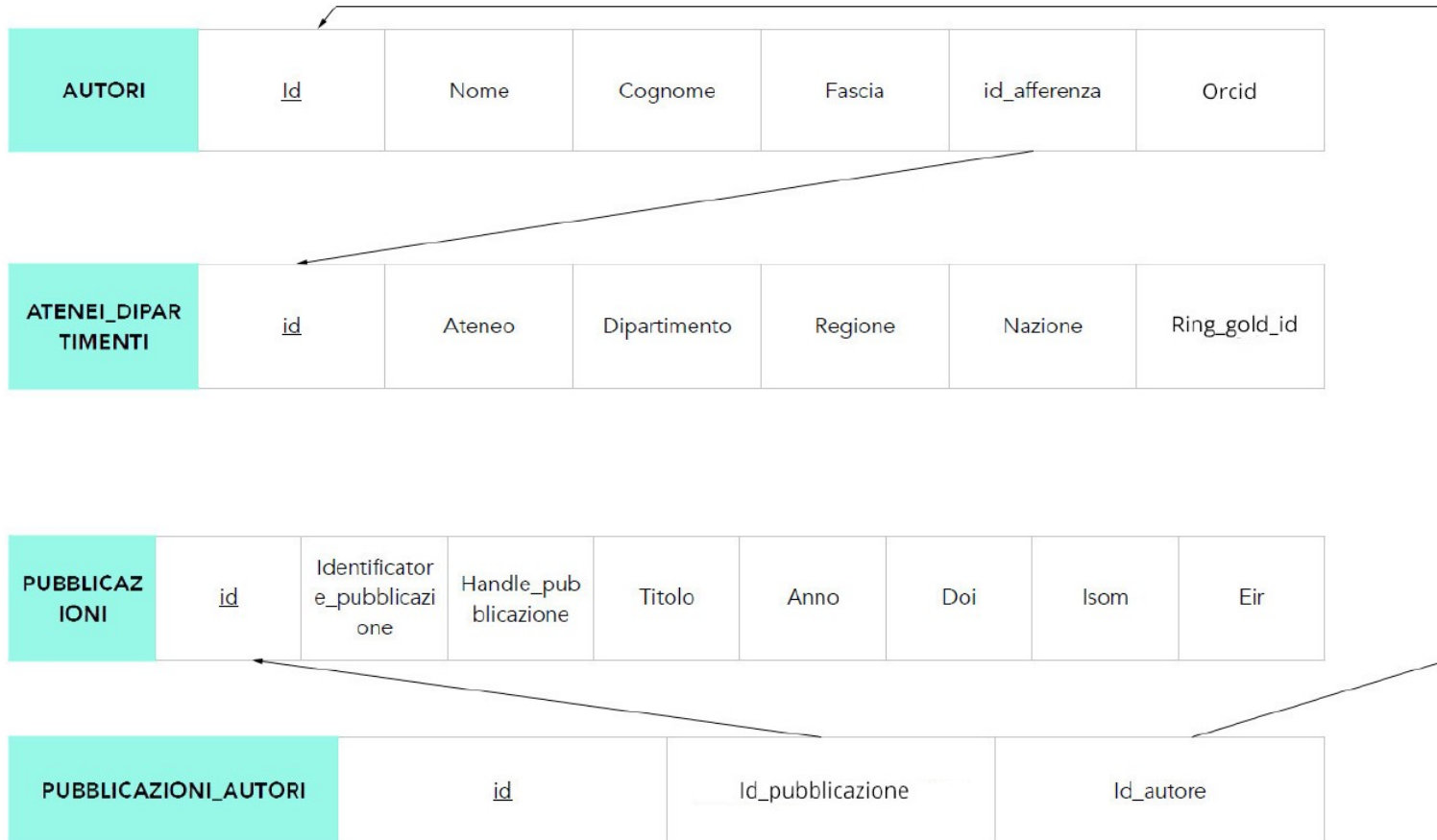
► Pubblicazioni

| Colonna | Tipo | Commento |
|-------------------------------------|---------------|----------|
| Id | int(11) | |
| Identificatore_pubblicazione | int(11) | |
| Handle_pubblicazione | varchar(1000) | |
| Titolo | varchar(1000) | |
| Anno | int(11) | |
| Doi | varchar(1000) | |
| Issn | varchar(1000) | |
| Eir | varchar(1000) | |

► Pubblicazioni_autori

| Colonna | Tipo | Commento |
|-------------------------|---------|----------|
| Id | int(11) | |
| Id_pubblicazione | int(11) | |
| Id_autore | int(11) | |

Vincoli di referenzialità



| Colonna | Tipo | Commento |
|---------------------|---------------|----------|
| Id | int(11) | |
| Nome | varchar(1000) | |
| Cognome | varchar(1000) | |
| Fascia | varchar(1000) | |
| Id_afferenza | int(11) | |
| Orcid | varchar(1000) | |

Chiavi esterne

| Sorgente | Obiettivo | ON DELETE | ON UPDATE | |
|---------------------|-------------------------|-----------|-----------|--------------------------|
| Id_afferenza | Atenei_dipartimenti(Id) | CASCADE | NO ACTION | Modifica |

ON DELETE: ON UPDATE:

Salva

RESTRICT
NO ACTION
CASCADE
SET NULL
SET DEFAULT

Indici

- ▶ Un indice è una struttura dati ausiliaria che consente di recuperare più velocemente i dati di una tabella, evitandone la lettura dell'intero contenuto (full table scan), tramite una selezione più mirata. L'indice può essere assegnato anche ad uno o più campi.
- ▶ L'importanza degli indici ed il loro impatto sulle **performances** è maggiore all'aumentare delle dimensioni della tabella: se in tabelle di pochi record l'aggiunta di indici può considerarsi "facoltativa", in tabelle con milioni di record la previsione di uno o più indici diventa assolutamente indispensabile in quanto, in mancanza, **l'engine** sarebbe chiamato ad un enorme sforzo ad ogni operazione di selezione dati.
- ▶ Prima di definire su quali campi creare gli indici occorre valutare quali siano le operazioni di selezione più frequenti. La giusta scelta degli indici in uno schema può migliorare le prestazioni dell'operazione di lettura anche dell'80%.

Creazione Indici

- Gli indici possono essere creati durante la creazione della tabella.

```
create table Autori (Id int,  
                    Nome varchar(1000),  
                    Cognome varchar(1000),  
                    Fascia varchar(1000),  
                    Id_afferenza int,  
                    Orcid varchar(1000),  
                    primary key (Id),  
                    foreign key (Id_afferenza) references Atenei_dipartimenti (Id),  
                    index index_cognome (Cognome));
```

| Colonna | Tipo | Commento |
|---------------------|---------------|----------|
| Id | int(11) | |
| Nome | varchar(1000) | |
| Cognome | varchar(1000) | |
| Fascia | varchar(1000) | |
| Id_afferenza | int(11) | |
| Orcid | varchar(1000) | |

- Possono essere aggiunti successivamente.

```
create index index_cognome on Autori (Cognome)
```

Indici

| | |
|----------------|---------------------|
| PRIMARY | <i>Id</i> |
| INDEX | <i>Id_afferenza</i> |
| INDEX | <i>Cognome(255)</i> |

Modifica indici

Chiavi esterne

| Sorgente | Obiettivo | ON DELETE | ON UPDATE | |
|----------------------------|---------------------------------------|-----------|-----------|----------|
| <i>Id_afferenza</i> | <i>Atenei_dipartimenti(Id)</i> | CASCADE | NO ACTION | Modifica |

- Possono essere eliminati.

```
drop index index_cognome on Autori
```


Creazione Indici

► Query su tabella senza indice

```
select * from Autori where Cognome = "Bistarelli"
```

| Id | Nome | Cognome | Fascia | Id_afferenza | Orcid |
|-----|---------|------------|----------------------|--------------|-------|
| 110 | Stefano | BISTARELLI | Associato confermato | 627 | 0 |

1 riga (0.042 s) [Modifica](#), [EXPLAIN](#), [Esporta](#)

► Query su tabella con indice

```
select * from Autori where Cognome = "Bistarelli"
```

| Id | Nome | Cognome | Fascia | Id_afferenza | Orcid |
|-----|---------|------------|----------------------|--------------|-------|
| 110 | Stefano | BISTARELLI | Associato confermato | 627 | 0 |

1 riga (0.001 s) [Modifica](#), [EXPLAIN](#), [Esporta](#)