

Esempio di database relazionale con l'utilizzo del prodotto MySQL

Marco Liverani*

Aprile 2015

In queste pagine viene riportato in sintesi il progetto di un database relazionale esemplificativo con cui viene rappresentato un modello informativo relativo agli studenti iscritti ad una Università. Il modello permette di rappresentare le informazioni relative agli studenti, ai corsi, ai corsi di laurea attivi presso diversi dipartimenti e la registrazione dell'esito degli esami degli studenti.

1 Schema Entità/Relazioni del modello

L'entità fondamentale del modello è costituito dall'entità **Studente**, che è caratterizzata dagli attributi anagrafici di ciascuno studente iscritto ad un Corso di Laurea dell'Ateneo.

Il **Dipartimento** è un'altra delle entità del modello e rappresenta i diversi dipartimenti presenti nell'Ateneo; ciascun Dipartimento appartiene ad una delle 14 aree scientifiche definite nell'ordinamento universitario italiano (es.: Area 01 "Scienze matematiche e informatiche", Area 02 "Scienze fisiche", ecc.). Ogni Dipartimento dell'Università ha un'offerta didattica costituita da uno o più Corsi di Laurea, rappresentati nel modello dall'entità **Corso di laurea**. Ogni studente è iscritto al più ad un Corso di laurea.

Gli istituti scolastici (licei, istituti tecnici, ecc.) sono rappresentati dall'entità **Scuola**. Esiste una relazione tra l'entità **Studente** e l'entità **Scuola** rappresentata dal **Titolo di studio** conseguito per potersi iscrivere all'Università; la relazione è caratterizzata da due attributi: l'anno di conseguimento del titolo di studio e il voto dell'esame di maturità.

Ciascuno studente possiede delle competenze linguistiche caratterizzate da un livello di comprensione della lingua straniera parlata e scritta, rappresentate nel modello dalla relazione **Capacità linguistica** tra l'entità **Studente** e l'entità **Lingua**, con cui vengono rappresentate le lingue straniere conosciute dagli studenti iscritti all'Ateneo.

Nel modello sono rappresentati i docenti dell'Ateneo, rappresentati dall'entità **Docente**, caratterizzata da attributi di carattere anagrafico; l'entità **Docente** è in relazione con l'entità **Corso**, che rappresenta uno dei corsi di cui il docente è titolare. Tra l'entità **Corso** e l'entità **Studente** è presente la relazione **Esame** che rappresenta gli esami sostenuti dagli studenti iscritti ad un Corso di laurea dell'Ateneo; la relazione **Esame** è caratterizzata dagli attributi relativi al voto con cui lo studente ha superato l'esame e l'anno accademico in cui tale esame è stato sostenuto.

In Figura 1 è rappresentato il modello Entità/Relazioni descritto nelle righe precedenti. Il modello potrebbe essere arricchito da ulteriori relazioni tra le entità: ad esempio una relazione tra i corsi e i

*E-mail: liverani@mat.uniroma3.it; ultima modifica: 25 aprile 2015

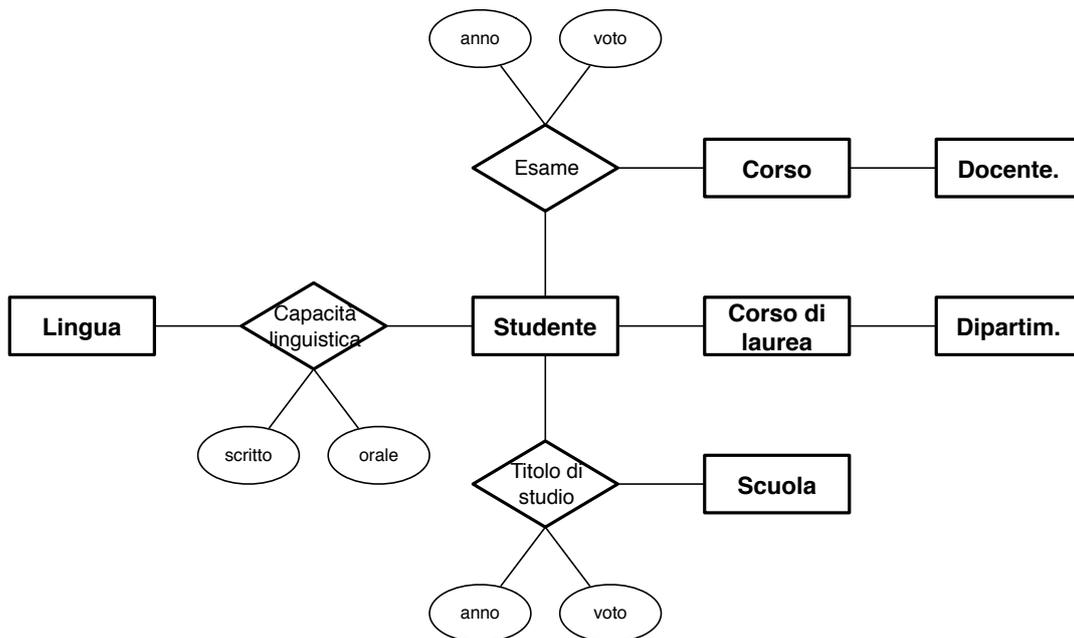


Figura 1: Schema entità/relazioni del database “Studenti”

Corsi di laurea (tenendo conto che però uno stesso corso potrebbe essere presente nell’offerta didattica di più Corsi di laurea) ed una relazione tra i docenti e i corsi, rendendo possibile l’attivazione di più “canali” di uno stesso corso, tenuti da docenti diversi.

2 Schema fisico del modello

Lo schema fisico delle tabelle e delle relazioni costruite mediante le chiavi esterne tra le tabelle, è rappresentato in Figura 2. Lo schema è stato costruito utilizzando il prodotto Open Source “MySQL Workbench” *community edition*.

I campi chiave sono rappresentati con una piccola chiave di colore giallo sulla sinistra; per le chiavi esterne (*foreign key*) la chiave è invece rappresentata di colore rosso. I campi caratterizzati dal vincolo *not null* sono rappresentati con un rombo di colore azzurro sulla sinistra.

Accanto ad ogni attributo delle tabelle è riportato il tipo di dato (nella sintassi di MySQL) utilizzato per definirlo.

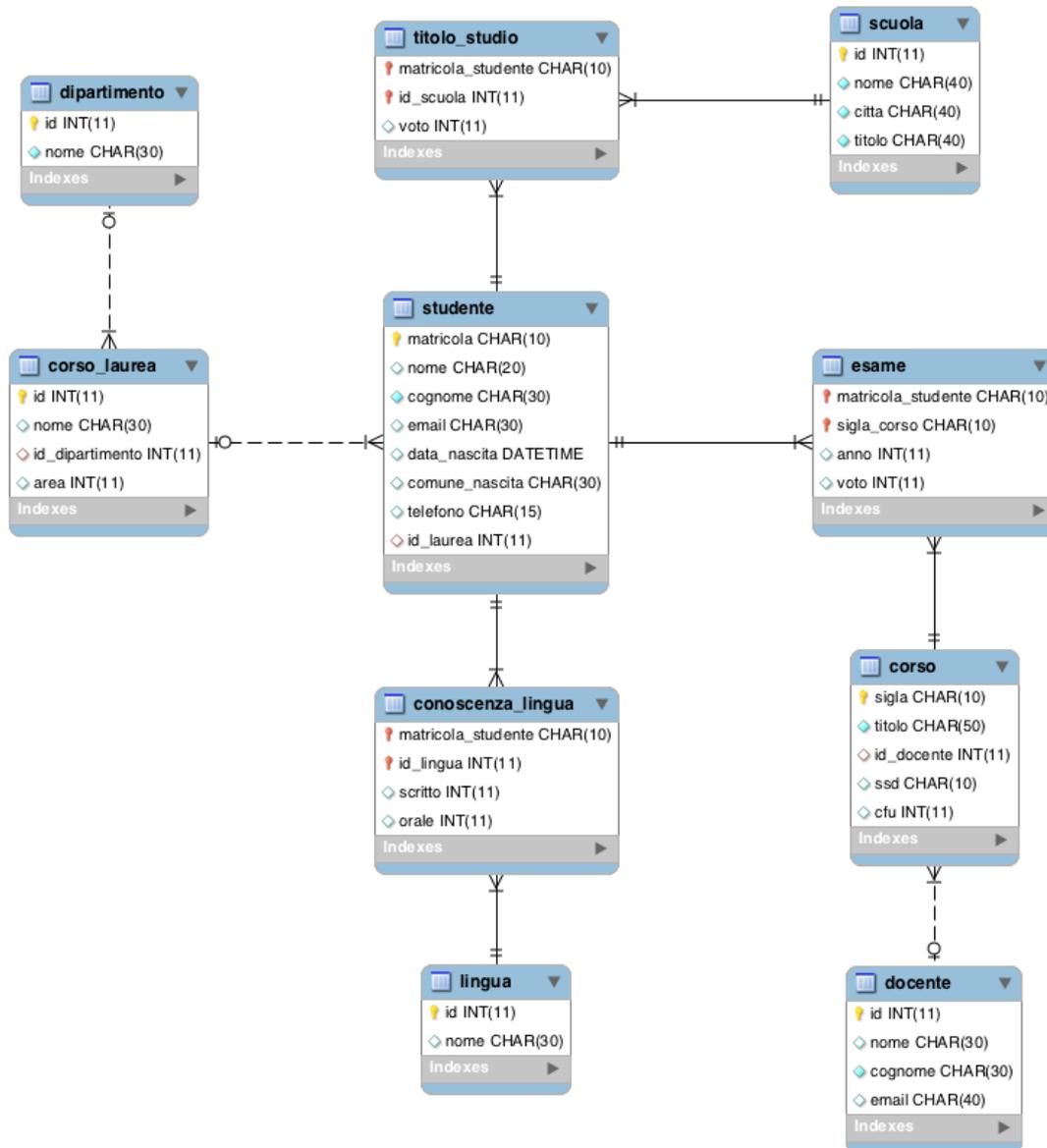


Figura 2: Schema fisico delle tabelle del database “Studenti”

3 Creazione della base dati

Di seguito sono riportate le istruzioni in linguaggio SQL per la creazione del database, delle tabelle e dei vincoli di integrità referenziale tra le tabelle del database. La sintassi utilizzata è quella del prodotto MySQL. L'indicazione "engine=innodb", tipica di MySQL, riportata nell'istruzione di creazione di ogni tabella, consente di utilizzare una rappresentazione fisica degli oggetti che supporta l'integrità referenziale.

```
1  --
2  -- DATABASE DEGLI STUDENTI
3  --
4  -- Script SQL per la creazione del database, delle tabelle e delle relazioni.
5  --
6  -- #(@)20150419(liverani@mat.uniroma3.it)
7  --
8
9  --
10 -- Creazione del database STUDENTI
11 --
12 create database studenti;
13 use studenti;
14
15 --
16 -- Creazione tabella DIPARTIMENTO
17 --
18 create table dipartimento (
19     id int not null auto_increment primary key,
20     nome char(30) not null
21 ) engine=innodb;
22
23 --
24 -- Creazione tabella CORSO_LAUREA
25 --
26 create table corso_laurea (
27     id int not null auto_increment primary key,
28     nome char(30),
29     id_dipartimento int,
30     area int,
31     foreign key (id_dipartimento) references dipartimento(id)
32 ) engine=innodb;
33
34 --
35 -- Creazione tabella STUDENTE
36 --
37 create table studente (
38     matricola char(10) not null primary key,
39     nome char(20),
```

```

40 | cognome char(30) not null,
41 | email char(30),
42 | data_nascita datetime,
43 | comune_nascita char(30),
44 | telefono char(15),
45 | id_laurea int,
46 | foreign key (id_laurea) references corso_laurea(id)
47 | ) engine=innodb;
48 |
49 | --
50 | -- Creazione tabella DOCENTE
51 | --
52 | create table docente (
53 |   id int not null auto_increment primary key,
54 |   nome char(30),
55 |   cognome char(30) not null,
56 |   email char(40)
57 | ) engine=innodb;
58 |
59 | --
60 | -- Creazione tabella CORSO
61 | --
62 | create table corso (
63 |   sigla char(10) not null primary key,
64 |   titolo char(50) not null,
65 |   id_docente int,
66 |   ssd char(10),
67 |   cfu int,
68 |   foreign key (id_docente) references docente(id)
69 | ) engine=innodb;
70 |
71 | --
72 | -- Creazione tabella ESAME
73 | --
74 | create table esame (
75 |   matricola_studente char(10) not null,
76 |   sigla_corso char(10) not null,
77 |   anno int,
78 |   voto int,
79 |   primary key (matricola_studente, sigla_corso),
80 |   foreign key (matricola_studente) references studente(matricola)
81 |   on update cascade on delete cascade,
82 |   foreign key (sigla_corso) references corso(sigla) on update cascade on delete restrict
83 | ) engine=innodb;
84 |
85 | --
86 | -- Creazione tabella LINGUA

```

```

87  --
88  create table lingua (
89      id int not null auto_increment primary key,
90      nome char(30)
91  ) engine=innodb;
92
93  --
94  -- Creazione tabella CONOSCENZA_LINGUA
95  --
96  create table conoscenza_lingua (
97      matricola_studente char(10) not null,
98      id_lingua int,
99      scritto int,
100     orale int,
101     primary key (matricola_studente, id_lingua),
102     foreign key (matricola_studente) references studente(matricola),
103     foreign key (id_lingua) references lingua(id)
104 ) engine=innodb;
105
106  --
107  -- Creazione tabella SCUOLA
108  --
109  create table scuola (
110     id int not null auto_increment primary key,
111     nome char(40) not null,
112     citta char(40) not null,
113     titolo char(40) not null
114 ) engine=innodb;
115
116  --
117  -- Creazione tabella TITOLO_STUDIO
118  --
119  create table titolo_studio (
120     matricola_studente char(10) not null,
121     id_scuola int,
122     voto int,
123     primary key (matricola_studente, id_scuola),
124     foreign key (matricola_studente) references studente(matricola),
125     foreign key (id_scuola) references scuola(id)
126 ) engine=innodb;

```

4 Inserimento di dati di configurazione

Di seguito sono riportate le istruzioni SQL, nella sintassi adottata da MySQL, che permettono di inserire dei dati nelle varie tabelle del database. Queste istruzioni sono riportate come supporto per la comprensione del significato di ciascun campo delle tabelle, ma anche per costituire una base di esempio per l'esecuzione delle query presentate nella sezione successiva.

```
1  --
2  -- DATABASE DEGLI STUDENTI
3  --
4  -- Inserimento dati nelle tabelle.
5  --
6  -- #(@)20150419(liverani@mat.uniroma3.it)
7  --
8
9  --
10 -- Inserimento dati Dipartimenti
11 --
12 insert into dipartimento (id, nome) values (1, 'Matematica e Fisica');
13 insert into dipartimento (id, nome) values (2, 'Scienze della formazione');
14 insert into dipartimento (id, nome) values (3, 'Ingegneria');
15 insert into dipartimento (id, nome) values (4, 'Lettere e Filosofia');
16 insert into dipartimento (id, nome) values (5, 'Economia e Commercio');
17 insert into dipartimento (id, nome) values (6, 'Architettura');
18
19 --
20 -- Inserimento dati Corso di laurea
21 --
22 insert into corso_laurea (id, nome, id_dipartimento, area) values (1, 'Matematica', 1, 1);
23 insert into corso_laurea (id, nome, id_dipartimento, area) values (2, 'Fisica', 1, 2);
24 insert into corso_laurea (id, nome, id_dipartimento, area) values (3, 'Scienze formazione', 2, 11);
25 insert into corso_laurea (id, nome, id_dipartimento, area) values (4, 'Ingegneria meccanica', 3, 9);
26 insert into corso_laurea (id, nome, id_dipartimento, area) values (5, 'Ingegneria informatica', 3, 9);
27 insert into corso_laurea (id, nome, id_dipartimento, area) values (6, 'Ingegneria gestionale', 3, 9);
28 insert into corso_laurea (id, nome, id_dipartimento, area) values (7, 'Letteratura italiana', 4, 10);
29 insert into corso_laurea (id, nome, id_dipartimento, area) values (8, 'Storia Vicino Oriente', 4, 10);
30 insert into corso_laurea (id, nome, id_dipartimento, area) values (9, 'Storia moderna', 4, 10);
31 insert into corso_laurea (id, nome, id_dipartimento, area) values (10, 'Economia', 5, 13);
32 insert into corso_laurea (id, nome, id_dipartimento, area) values (11, 'Architettura', 6, 8);
33 insert into corso_laurea (id, nome, id_dipartimento, area) values (12, 'Restauro', 6, 8);
34
35 --
36 -- Inserimento dati Scuola
37 --
38 insert into scuola (id, nome, citta, titolo) values
39 (1, 'Liceo Scientifico "Peano"', 'Roma', 'Maturita" scientifica');
40 insert into scuola (id, nome, citta, titolo) values
```

```

41 (2, 'Liceo Scientifico "Morgagni"', 'Roma', 'Maturita" scientifica');
42 insert into scuola (id, nome, citta, titolo) values
43 (3, 'Liceo Classico "Virgilio"', 'Roma', 'Maturita" classica');
44 insert into scuola (id, nome, citta, titolo) values (4, 'ITIS "Armellini"', 'Roma', 'Perito industriale');
45 insert into scuola (id, nome, citta, titolo) values (5, 'ITIS "Fermi"', 'Roma', 'Perito industriale');
46
47 --
48 -- Inserimento dati Docente
49 --
50 insert into docente (id, nome, cognome, email) values (1, 'Marco', 'Liverani', 'liverani@uniroma3.it');
51 insert into docente (id, nome, cognome, email) values (2, 'Marco', 'Pedicini', 'pedicini@uniroma3.it');
52 insert into docente (id, nome, cognome, email) values (3, 'Gianni', 'Mancini', 'gianni@uniroma3.it');
53 insert into docente (id, nome, cognome, email) values (4, 'Marco', 'Fontana', 'fontana@uniroma3.it');
54 insert into docente (id, nome, cognome, email) values (5, 'Roberto', 'Ferretti', 'ferretti@uniroma3.it');
55 insert into docente (id, nome, cognome, email) values (6, 'Michele', 'Abrusci', 'abrusci@uniroma3.it');
56
57 --
58 -- Inserimento dati Corso
59 --
60 insert into corso (sigla, titolo, id_docente, ssd, cfu) values ('IN110', 'Informatica 1', 1, 'INF/01', 10);
61 insert into corso (sigla, titolo, id_docente, ssd, cfu) values ('AL110', 'Algebra 1', 4, 'MAT/02', 10);
62 insert into corso (sigla, titolo, id_docente, ssd, cfu) values ('LM410', 'Logica 1', 6, 'MAT/01', 7);
63 insert into corso (sigla, titolo, id_docente, ssd, cfu) values ('IN440', 'Ottim. Combin.', 1, 'INF/01', 7);
64 insert into corso (sigla, titolo, id_docente, ssd, cfu) values ('IN530', 'Sist.Elabor.Inform.', 1, 'INF/01', 4);
65 insert into corso (sigla, titolo, id_docente, ssd, cfu) values ('AM210', 'Analisi Mat. 3', 3, 'MAT/05', 9);
66
67 --
68 -- Inserimento dati Lingue
69 --
70 insert into lingua (id, nome) values (1, 'Inglese');
71 insert into lingua (id, nome) values (2, 'Francese');
72 insert into lingua (id, nome) values (3, 'Spagnolo');
73 insert into lingua (id, nome) values (4, 'Tedesco');
74 insert into lingua (id, nome) values (5, 'Cinese');
75 insert into lingua (id, nome) values (6, 'Portoghese');
76
77 --
78 -- Inserimento dati studenti
79 --
80 insert into studente (matricola, nome, cognome, email, data_nascita,
81 comune_nascita, telefono, id_laurea) values ('101010', 'Giovanni',
82 'Rossi', 'giovannired@gmail.com', '1993-04-21', 'Roma', '335-123456', 1);
83 insert into studente (matricola, nome, cognome, email, data_nascita,
84 comune_nascita, telefono, id_laurea) values ('202020', 'Giulia', 'Verdi',
85 'giulia94@yahoo.com', '1994-11-05', 'Napoli', '06-907632', 1);
86 insert into studente (matricola, nome, cognome, email, data_nascita,
87 comune_nascita, telefono, id_laurea) values ('303030', 'Chiara', 'Verdi',

```

```

88 'chiaretta@libero.it', '1992-01-15', 'Napoli', '333-344556', 1);
89 insert into studente (matricola, nome, cognome, email, data_nascita,
90 comune_nascita, telefono, id_laurea) values ('404040', 'Nicola', 'Leone',
91 'n.leone@libero.it', '1995-03-28', 'Roma', '333-102938', 1);
92 insert into studente (matricola, nome, cognome, email, data_nascita,
93 comune_nascita, telefono, id_laurea) values ('505050', 'Mario',
94 'Dell"Acqua', 'dellacqua@gmail.com', '1994-06-20', 'Roma', '335-918273', 1);
95
96 --
97 -- Inserimento dati Titolo di studio
98 --
99 insert into titolo_studio (matricola_studente, id_scuola, voto) values ('101010', 1, 100);
100 insert into titolo_studio (matricola_studente, id_scuola, voto) values ('202020', 2, 90);
101 insert into titolo_studio (matricola_studente, id_scuola, voto) values ('303030', 1, 87);
102 insert into titolo_studio (matricola_studente, id_scuola, voto) values ('404040', 4, 100);
103 insert into titolo_studio (matricola_studente, id_scuola, voto) values ('505050', 5, 85);
104
105 --
106 -- Inserimento dati sulle Conoscenze linguistiche
107 --
108 insert into conoscenza_lingua (matricola_studente, id_lingua, scritto, orale) values ('101010', 1, 8, 7);
109 insert into conoscenza_lingua (matricola_studente, id_lingua, scritto, orale) values ('101010', 3, 7, 5);
110 insert into conoscenza_lingua (matricola_studente, id_lingua, scritto, orale) values ('202020', 1, 6, 6);
111 insert into conoscenza_lingua (matricola_studente, id_lingua, scritto, orale) values ('303030', 1, 5, 6);
112 insert into conoscenza_lingua (matricola_studente, id_lingua, scritto, orale) values ('303030', 2, 7, 7);
113 insert into conoscenza_lingua (matricola_studente, id_lingua, scritto, orale) values ('404040', 3, 8, 8);
114
115 --
116 -- Inserimento dati sugli Esami
117 --
118 insert into esame (matricola_studente, sigla_corso, anno, voto) values ('101010', 'IN110', 2013, 30);
119 insert into esame (matricola_studente, sigla_corso, anno, voto) values ('101010', 'AL110', 2013, 28);
120 insert into esame (matricola_studente, sigla_corso, anno, voto) values ('101010', 'IN440', 2014, 28);
121 insert into esame (matricola_studente, sigla_corso, anno, voto) values ('202020', 'IN110', 2012, 22);
122 insert into esame (matricola_studente, sigla_corso, anno, voto) values ('202020', 'IN440', 2014, 26);
123 insert into esame (matricola_studente, sigla_corso, anno, voto) values ('202020', 'AM210', 2014, 30);
124 insert into esame (matricola_studente, sigla_corso, anno, voto) values ('202020', 'LM410', 2014, 27);
125 insert into esame (matricola_studente, sigla_corso, anno, voto) values ('303030', 'IN110', 2012, 24);
126 insert into esame (matricola_studente, sigla_corso, anno, voto) values ('404040', 'IN110', 2013, 29);

```

5 Alcune query di esempio

5.1 Selezione da dati presenti in una sola tabella

Selezione di tutti i dati presenti in una determinata tabella; ad esempio, selezione di tutti i Corsi di laurea attivi nell'Ateneo:

```
mysql> select * from corso_laurea;
+-----+-----+-----+-----+
| id | nome                               | id_dipartimento | area |
+-----+-----+-----+-----+
| 1 | Matematica                         | 1               | 1    |
| 2 | Fisica                             | 1               | 2    |
| 3 | Scienze formazione primaria       | 2               | 11   |
| 4 | Ingegneria meccanica              | 3               | 9    |
| 5 | Ingegneria informatica            | 3               | 9    |
| 6 | Ingegneria gestionale              | 3               | 9    |
| 7 | Letteratura italiana              | 4               | 10   |
| 8 | Storia Vicino Oriente              | 4               | 10   |
| 9 | Storia moderna                    | 4               | 10   |
| 10 | Economia                          | 5               | 13   |
| 11 | Architettura                     | 6               | 8    |
| 12 | Restauro                          | 6               | 8    |
+-----+-----+-----+-----+
```

Selezione di alcune colonne di una tabella, ad esempio selezione di alcuni degli attributi della tabella Studente:

```
mysql> select nome, cognome from studente order by cognome;
+-----+-----+
| nome   | cognome |
+-----+-----+
| Mario  | Dell'Acqua |
| Nicola | Leone    |
| Giovanni | Rossi   |
| Giulia | Verdi    |
| Chiara | Verdi    |
+-----+-----+
```

Selezione di alcuni attributi di una tabella e solo di alcune righe, quelle che corrispondono alla condizione indicata nella query; ad esempio selezioniamo solo i corsi dei settori scientifico disciplinari della Matematica (sono identificati dalle sigle "Mat/01", "Mat/02", ecc.):

```
mysql> select * from corso;
+-----+-----+-----+-----+-----+
| sigla | titolo                               | id_docente | ssd   | cfu |
+-----+-----+-----+-----+-----+
| AL110 | Algebra 1                           | 4          | MAT/02 | 10 |
| AM210 | Analisi Matematica 3                 | 3          | MAT/05 | 9  |
| IN110 | Informatica 1                        | 1          | INF/01 | 10 |
| IN440 | Ottimizzazione Combinatoria         | 1          | INF/01 | 7  |
| IN530 | Sist.Elabor.Informazioni             | 1          | INF/01 | 4  |
| LM410 | Logica 1                             | 6          | MAT/01 | 7  |
+-----+-----+-----+-----+-----+
```

```
mysql> select sigla, titolo, cfu from corso where ssid like 'MAT%';
+-----+-----+-----+
| sigla | titolo          | cfu |
+-----+-----+-----+
| AL110 | Algebra 1      | 10 |
| AM210 | Analisi Matematica 3 | 9 |
| LM410 | Logica 1       | 7 |
+-----+-----+-----+
```

L'operatore `like` consente di verificare se una stringa di caratteri corrisponde parzialmente a quella indicata tra apici, utilizzando anche il simbolo “%”, che rappresenta il carattere *wildcard* (il simbolo che sostituisce un carattere o una stringa qualsiasi di caratteri) in linguaggio SQL.

Una tabella potrebbe presentare valori ripetuti in due o più record; con l'operatore `distinct` è possibile selezionare solo le “copie uniche” delle righe prodotte mediante un'istruzione `select`:

```
mysql> select ssid from corso order by ssid;
+-----+
| ssid |
+-----+
| INF/01 |
| INF/01 |
| INF/01 |
| MAT/01 |
| MAT/02 |
| MAT/05 |
+-----+
6 rows in set (0.00 sec)
```

```
mysql> select distinct ssid from corso order by ssid;
+-----+
| ssid |
+-----+
| INF/01 |
| MAT/01 |
| MAT/02 |
| MAT/05 |
+-----+
4 rows in set (0.00 sec)
```

5.2 Join: selezione di dati ottenuti intersecando più tabelle

Con l'istruzione `select` può essere compiuta l'operazione di *join* tra più tabelle del database; tale operazione comporta l'esecuzione del prodotto cartesiano tra tutte le righe delle tabelle utilizzate nella *join*, selezionando poi tra le righe del prodotto cartesiano solo quelle per le quali risulta verificata la condizione espressa nell'istruzione `select`. Ad esempio, intersecando i dati presenti nelle tabelle `Studiante`, `Corso` ed `Esame`, selezioniamo i voti presi dagli studenti agli esami che hanno sostenuto:

```
mysql> select studente.nome, studente.cognome, corso.sigla, esame.voto
-> from studente, corso, esame
-> where studente.matricola=esame.matricola_studente
-> and esame.sigla_corso=corso.sigla
-> order by studente.cognome;
```

nome	cognome	sigla	voto
Nicola	Leone	IN110	29
Giovanni	Rossi	AL110	28
Giovanni	Rossi	IN110	30
Giovanni	Rossi	IN440	28
Giulia	Verdi	AM210	30
Giulia	Verdi	IN110	22
Giulia	Verdi	IN440	26
Giulia	Verdi	LM410	27
Chiara	Verdi	IN110	24

Selezioniamo tutti i dati relativi agli esami sostenuti dallo studente con matricola “101010”:

```
mysql> select studente.nome, studente.cognome, corso.sigla, corso.titolo,
-> esame.voto, corso.cfu
-> from studente, corso, esame
-> where studente.matricola=esame.matricola_studente
-> and esame.sigla_corso=corso.sigla and studente.matricola='101010';
```

nome	cognome	sigla	titolo	voto	cfu
Giovanni	Rossi	AL110	Algebra 1	28	10
Giovanni	Rossi	IN110	Informatica 1	30	10
Giovanni	Rossi	IN440	Ottimizzazione Combinatoria	28	7

5.3 Operazioni di gruppo

Le operazioni di gruppo sono funzioni matematiche elementari che consentono di eseguire il conteggio dei record, la somma o la media dei valori di determinati campi numerici. Tali funzioni devono necessariamente essere applicate ad un insieme di record selezionati mediante una `select`; il criterio di raggruppamento dei record selezionati, al fine di calcolare le funzioni di conteggio (`count`), sommatoria (`sum`) e di media aritmetica (`avg`) viene espresso mediante la clausola `group by` dell’istruzione `select`.

Ad esempio contiamo il numero di corsi presenti in archivio. Da notare l’opzione “`as`” che permette di assegnare un nome valido nell’ambito della query ad una delle colonne oggetto della query stessa.

```
mysql> select * from corso;
```

sigla	titolo	id_docente	ssd	cfu
AL110	Algebra 1	4	MAT/02	10
AM210	Analisi Matematica 3	3	MAT/05	9
IN110	Informatica 1	1	INF/01	10
IN440	Ottimizzazione Combinatoria	1	INF/01	7
IN530	Sist.Elabor.Informazioni	1	INF/01	4
LM410	Logica 1	6	MAT/01	7

6 rows in set (0.00 sec)

```
mysql> select count(*) as 'Numero corsi' from corso;
+-----+
| Numero corsi |
+-----+
|           6 |
+-----+
1 row in set (0.00 sec)
```

Le funzioni di gruppo possono essere applicate anche nell'ambito di un'operazione di *join*. Ad esempio contiamo il numero di corsi tenuti da ciascun docente:

```
mysql> select * from corso;
+-----+-----+-----+-----+-----+
| sigla | titolo                | id_docente | ssd   | cfu |
+-----+-----+-----+-----+-----+
| AL110 | Algebra 1             |           4 | MAT/02 | 10 |
| AM210 | Analisi Matematica 3  |           3 | MAT/05 | 9  |
| IN110 | Informatica 1         |           1 | INF/01 | 10 |
| IN440 | Ottimizzazione Combinatoria |           1 | INF/01 | 7  |
| IN530 | Sist.Elabor.Informazioni |           1 | INF/01 | 4  |
| LM410 | Logica 1              |           6 | MAT/01 | 7  |
+-----+-----+-----+-----+-----+
```

6 rows in set (0.00 sec)

```
mysql> select id_docente, count(*) as corsi from corso group by id_docente;
+-----+-----+
| id_docente | corsi |
+-----+-----+
|           1 |     3 |
|           3 |     1 |
|           4 |     1 |
|           6 |     1 |
+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select docente.cognome, count(*) as corsi
-> from docente, corso
-> where corso.id_docente=docente.id
-> group by id_docente;
+-----+-----+
| cognome | corsi |
+-----+-----+
| Liverani |     3 |
| Mancini  |     1 |
| Fontana  |     1 |
| Abrusci  |     1 |
+-----+-----+
4 rows in set (0.00 sec)
```

Con la funzione *avg* (*average*) si può calcolare la media aritmetica di un insieme di valori numerici. Calcoliamo la media dei voti degli studenti, presentandoli dal migliore (con la media più alta) al peggiore:

```
mysql> select studente.cognome, studente.nome, avg(esame.voto) as media
-> from studente, esame
-> where studente.matricola=esame.matricola_studente
-> group by esame.matricola_studente
-> order by media desc;
```

```
+-----+-----+-----+
| cognome | nome      | media  |
+-----+-----+-----+
| Leone   | Nicola   | 29.0000 |
| Rossi   | Giovanni | 28.6667 |
| Verdi   | Giulia   | 26.2500 |
| Verdi   | Chiara   | 24.0000 |
+-----+-----+-----+
```

Calcoliamo infine il numero di crediti (CFU) maturati dagli studenti sostenendo i diversi esami dei corsi. Per calcolare i crediti cumulati, utilizziamo la funzione `sum` che produce la sommatoria di un insieme di valori numerici:

```
mysql> select studente.nome, studente.cognome, sum(corso.cfu) as cfu
-> from studente, corso, esame
-> where studente.matricola=esame.matricola_studente
-> and corso.sigla=esame.sigla_corso
-> group by esame.matricola_studente
-> order by studente.cognome, studente.nome;
```

```
+-----+-----+-----+
| nome     | cognome  | cfu    |
+-----+-----+-----+
| Nicola   | Leone    | 10     |
| Giovanni | Rossi    | 27     |
| Chiara   | Verdi    | 10     |
| Giulia   | Verdi    | 33     |
+-----+-----+-----+
4 rows in set (0.00 sec)
```