

Progetto Basi di Dati e Conoscenza
Donato Francesco Pio Stanco
Corso di Laurea in Informatica
Matricola 0251690
A.A. 2019/2020

BLOG FOTOGRAFICO

Basi di Dati e Conoscenza
Donato Francesco Pio Stanco



Indice

<u>1-ANALISI DEI REQUISITI.....</u>	<u>2</u>
1.1-GLOSSARIO DEI TERMINI.....	3
<u>2-MODELLO LOGICO E FISICO.....</u>	<u>5</u>
2.1-SCHEMA E-R LOGICO.....	5
2.2-SCHEMA E-R FISICO.....	6
2.3-TABELLE E ATTRIBUTI.....	6
<u>3-CREAZIONE TABELLE.....</u>	<u>7</u>
<u>4-INSERIMENTI DI PROVA.....</u>	<u>10</u>
<u>5-QUERY.....</u>	<u>17</u>
<u>6-ALGEBRA E CALCOLO RELAZIONALE.....</u>	<u>23</u>
<u>7-PARTI ULTERIORI DEL DB.....</u>	<u>24</u>
7.1-TRIGGER.....	24
7.2-STORED PROCEDURE.....	26
7.3-VIEW.....	26
<u>8-MONGO DB.....</u>	<u>27</u>
8.1-COLLECTION.....	27
8.2-QUERY.....	31
8.3-TEMPI DI RISPOSTA.....	32

1 - Analisi dei Requisiti

Si vuole realizzare il database “Blog Fotografico” pensato per la condivisione di foto da parte di fotografi professionisti, dando così la possibilità di visionare i loro scatti agli utenti presenti all’interno di questo blog.

Per ogni foto pubblicata saranno visibili le seguenti informazioni:

- le informazioni relative ai vari parametri dello scatto, ad esempio il tempo d’esposizione, il valore degli ISO, l’apertura del diaframma e tanti altri;
- la macchina e l’obiettivo utilizzato;
- una breve descrizione delle foto, finalizzata a valorizzare lo scatto proposto;
- la categoria a cui lo stesso scatto appartiene, ovvero il genere fotografico della foto, ad esempio natura, ritratti, tramonto e tanti altri;
- la data in cui la foto è stata scatta (ad esempio 10-01-2020);
- l’indirizzo del luogo nel quale la foto è stata scattata, ad esempio la Via, il numero civico, la città, la provincia e altri;

Ogni utente, per ognuna delle foto presenti all’interno del blog, può assegnare un unico voto da 1 a 10 e insieme a questo, aggiungere un commento per esprimere il perché della propria scelta, dando un parere su quali parametri da cambiare per ottenere degli scatti migliori, oppure consigliare un’ottica o una macchina più adatte allo scopo o al contesto della foto in questione.

Gli utenti possono pubblicare più album fotografici e dare loro una breve descrizione, in particolare gli album possono contenere un numero variabile di foto, ma una foto può essere pubblicata in un unico album.

Ogni utente è identificato tramite un ID numerico, e deve inserire vari dati personali, ad esempio il nome, il cognome, la mail e altri parametri, inoltre è possibile che questo utente lavori per uno studio fotografico, la cosa importante è che un utente può lavorare solo per uno studio, i quali possono avere più sedi sparse per il paese, infatti nella base di dati sono memorizzate le informazioni su dove questi sono ubicati, ad esempio l’indirizzo e la città.

Nella base di dati sono inoltre contenute le informazioni relativi ai vari modelli di macchine fotografiche come la marca (Nikon, Sony, Leica e altre), la dimensione del sensore (in millimetri), il peso (in grammi) o il massimo numero di ISO che la macchina può raggiungere, in aggiunta vengono memorizzate anche le informazioni relative agli obiettivi (ad esempio la marca, il modello, il massimo valore dello zoom, se supporta l’autofocus o la stabilizzazione ottica dell’immagine e altre) utilizzati dagli utenti in coppia con una determinata macchina.

Le macchine fotografiche possono essere suddivise in due categorie principali, ovvero:

- Reflex: sono dispositivi che possono montare diversi tipi di ottica se hanno lo stesso tipo di attacco;
- Compatte: sono macchine che hanno un’ottica fissa che non può cambiare, spesso di dimensioni ridotte e quindi facili da trasportare.

1.1 - Glossario dei termini

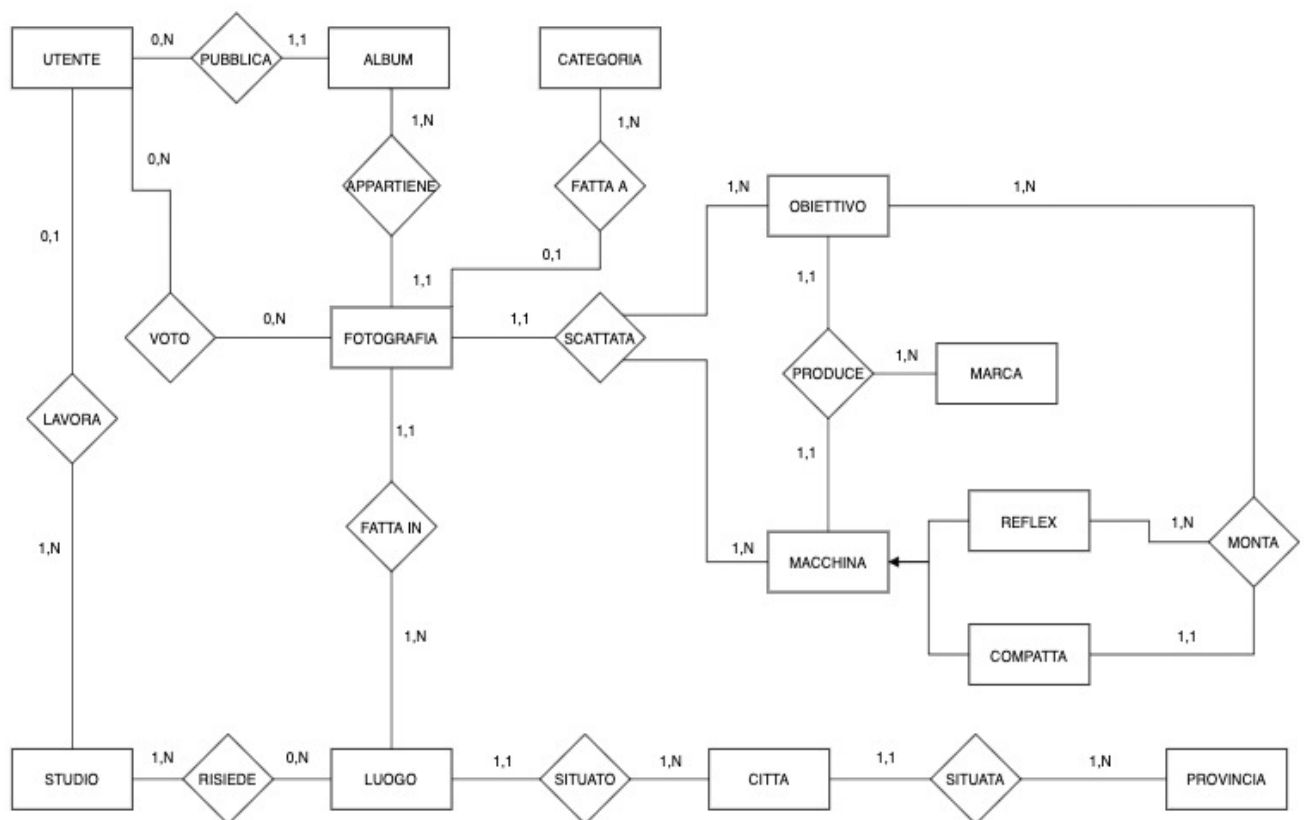
Termine	Descrizione	Collegamento
Utente	Colui che ha accesso al blog e può visualizzare e commentare delle foto pubblicate. Può anche pubblicare album.	Fotografia
Fotografia	Una fotografia è un'immagine statica ottenuta tramite un processo di registrazione permanente tra luce e materia. Una foto può essere pubblicata sul blog ed essere vista e commentata.	Utente, Macchina Fotografica
Macchina Fotografica	Apparecchio impiegato per effettuare la ripresa di fotografia. Generalmente portatile, con forma, dimensioni, struttura e caratteristiche tecniche diverse a secondo dell'uso cui è destinata	Utente, Fotografia
Macchina fotografica Reflex	La single-lens reflex (SLR), o più semplicemente reflex, è un tipo di fotocamera dotata di un sistema di mira che permette di osservare dal mirino ottico l'inquadratura in ingresso dallo stesso obiettivo. Inoltre possono montare diversi tipi di ottica se hanno lo stesso attacco.	Macchina Fotografica
Macchina Fotografica Compatta	La fotocamera compatta è una fotocamera non modulare e quindi con obiettivo non intercambiabile (sia zoom, che a focale fissa), spesso di dimensioni ridotte, quindi più facilmente portatile, ma costituite generalmente da una dotazione accessoriaria più completa (custodia, flash, mirino, ecc).	Macchina Fotografica

Blog	Sito web nel quale vengono pubblicati dei contenuti, fruibili agli utenti iscritti. I “blogger” sono coloro che periodicamente aggiornano i blog con i vari contenuti.	Utente
ISO	L’ISO è uno degli elementi che determinano la quantità di luce catturata dal sensore di una macchina fotografica digitale. Più precisamente, nella fotografia digitale, l’ISO indica la sensibilità del sensore della luce.	Macchina Fotografica
Obiettivo	Dispositivo ottico in grado di raccogliere e riprodurre un’immagine.	Macchina Fotografica
OIS	OIS (Optical Image Stabilisation) indica la stabilizzazione ottica di un apparecchio hardware usato per scattare foto.	Macchina Fotografica, Obiettivo
AF	AF (Autofocus) è un automatismo che applicato ai sistemi ottici permette di ottenere e mantenere automaticamente la messa a fuoco su un soggetto.	Obiettivo

2 - Modello Logico e Fisico

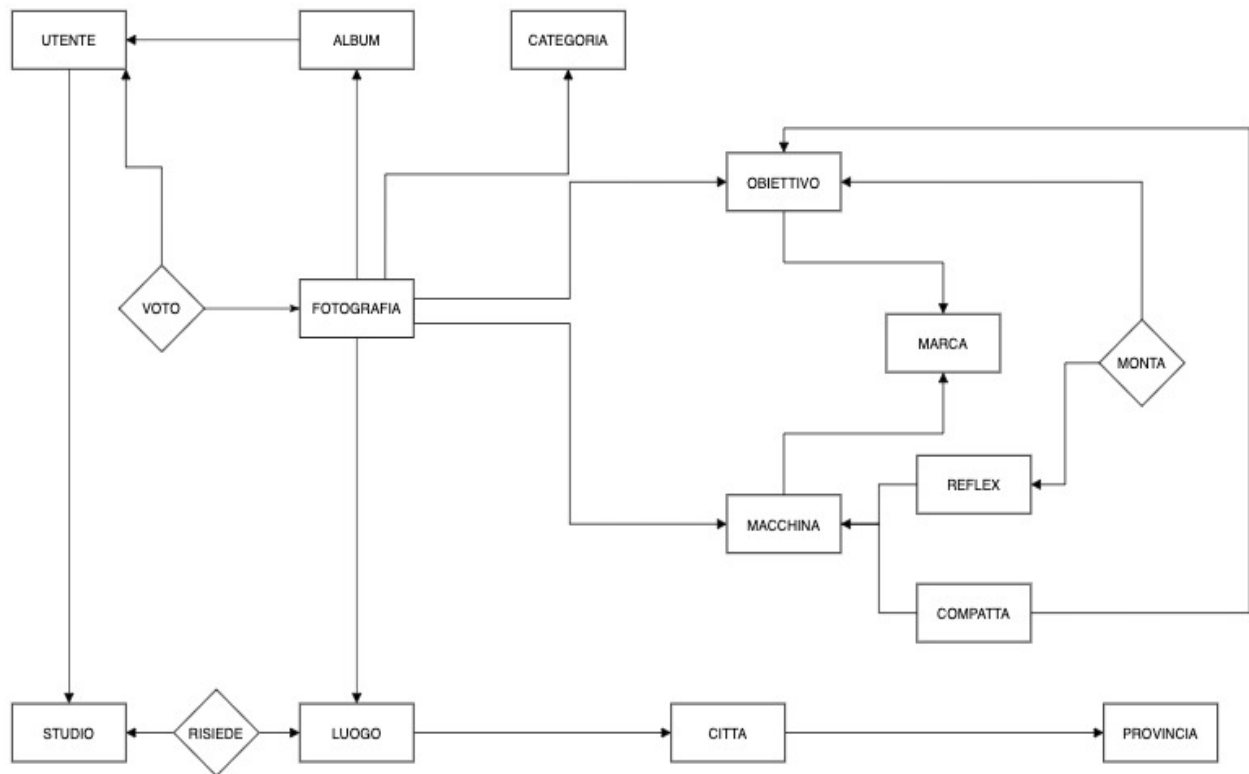
Progettare una base di dati significa definirne la struttura, le caratteristiche il contenuto attraverso l'uso di opportune metodologie.

2.1 - Schema E-R Logico



Nota: non ho inserito gli attributi direttamente nello schema E-R Logico per renderlo più leggibile.

2.2 - Schema E-R Fisico



Nota: non ho inserito chiavi primarie e attributi nello schema Fisico per renderlo più leggibili, per attributi e chiavi vedere paragrafo 2.3 del documento.

2.3 - Tabelle e Attributi

Per il database Blog Fotografico sono necessarie 16 tabelle, di cui 13 entità e 3 relazioni, ed ogni singola tabella è descritta nella lista sottostante.

Tabelle e Attributi:

PROVINCIA (ID, Nome, Sigla)
CITTA (ID, Nome, ID_Provincia (FK))
LUOGO (ID, Via, ID_Citta (FK))
STUDIO (ID, Nome)
RISIEDE (ID_Studio (FK), ID_Luogo (FK), NumeroCivico)
UTENTE (ID, Nome, Cognome, Data, Email, Cellulare, ID_Studio (FK))
ALBUM (ID, NomeAlbum, Data, ID_Utente (FK))
CATEGORIA (ID, Nome, Descrizione)
MARCA (ID, Nome, Stato)
MACCHINA (ID, Modello, Sensore, Peso, MaxISO, ID_Marca (FK))
OBIETTIVO (ID, Modello, OIS, AF, Zoom, ID_Marca (FK))
REFLEX (ID_Macchina (FK), MotoreAF)
COMPATTA (ID_Macchina (FK), ID_Obiettivo (FK), ZoomDigitale, StabilizzazioneDigitale)
MONTA (ID_Macchina (FK), ID_Obiettivo (FK), Attacco)
VOTO (ID, ID_Utente (FK), ID_Foto (FK), Voto, Descrizione)

FOTOGRAFIA (**ID**, Descrizione, Data, ID_Album (FK), ID_Categoria (FK), ID_Luogo (FK), ID_Macchina (FK), ID_Obiettivo (FK), LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO, Risoluzione)

Nota: Gli attributi in grassetto e sottolineati sono le chiavi primarie, mentre gli attributi seguiti dalla sigla FK sono le chiavi esterne.

3 - Creazione Tabelle

Creazione delle tabelle del database su MySQL tramite la parola chiave "create table".

Nota: Per facilitare e diminuire i tempi di elaborazione per ogni entità ho utilizzato un ID numerico incrementabile come chiave primaria.

```
create table PROVINCIA(  
    ID int not null AUTO_INCREMENT,  
    Nome varchar(50) not null,  
    Sigla varchar(2) not null,  
    primary key (ID)  
) engine=INNODB;  
  
create table CITTA(  
    ID int not null AUTO_INCREMENT,  
    Nome varchar(50) not null,  
    ID_Provincia int not null,  
    primary key (ID),  
    foreign key (ID_Provincia) references PROVINCIA (ID)  
) engine=INNODB;  
  
create table LUOGO(  
    ID int not null AUTO_INCREMENT,  
    Via varchar(50) not null,  
    ID_Citta int not null,  
    primary key (ID),  
    foreign key (ID_Citta) references CITTA (ID)  
) engine=INNODB;  
  
create table STUDIO(  
    ID int not null AUTO_INCREMENT,  
    Nome varchar(50) not null,  
    primary key (ID)  
) engine=INNODB;  
  
create table RISIEDE(  
    ID_Studio int not null,  
    ID_Luogo int not null,  
    NumeroCivico int not null,  
    primary key (ID_Studio, ID_Luogo),  
    foreign key (ID_Studio) references STUDIO (ID),  
    foreign key (ID_Luogo) references LUOGO (ID)  
) engine=INNODB;  
  
create table UTENTE(  
    ID int not null AUTO_INCREMENT,  
    Nome varchar(50) not null,  
    Cognome varchar(50) not null,  
    Data date,
```



```

        Email varchar(50) not null,
        Cellulare varchar(50) not null,
        ID_Studio int not null,
        primary key (ID),
        foreign key (ID_Studio) references STUDIO (ID)
    ) engine=INNODB;

create table ALBUM(
    ID int not null AUTO_INCREMENT,
    NomeAlbum varchar(50) not null,
    Data date,
    ID_Utente int not null,
    primary key (ID),
    foreign key (ID_Utente) references UTENTE (ID)
) engine=INNODB;

create table CATEGORIA(
    ID int not null AUTO_INCREMENT,
    Nome varchar(50) not null,
    Descrizione varchar(50) not null,
    primary key (ID)
) engine=INNODB;

create table MARCA(
    ID int not null AUTO_INCREMENT,
    Nome varchar(50) not null,
    Stato varchar(50) not null,
    primary key (ID)
) engine=INNODB;

create table MACCHINA(
    ID int not null AUTO_INCREMENT,
    Modello varchar(50) not null,
    Sensore int not null,
    Peso int not null,
    MaxISO int not null,
    ID_Marca int not null,
    primary key (ID),
    foreign key (ID_Marca) references MARCA (ID)
) engine=INNODB;

create table OBIETTIVO(
    ID int not null AUTO_INCREMENT,
    Modello varchar(50) not null,
    OIS bit not null,
    AF bit not null,
    Zoom int not null,
    ID_Marca int not null,
    primary key (ID),
    foreign key (ID_Marca) references MARCA (ID)
) engine=INNODB;

create table COMPATTA(
    ID_Macchina int not null,
    ID_Obiettivo int not null,
    ZoomDigitale int not null,
    StabilizzazioneDigitale bit not null,
    primary key (ID_Macchina, ID_Obiettivo),
    foreign key (ID_Macchina) references MACCHINA (ID),
    foreign key (ID_Obiettivo) references OBIETTIVO (ID)
) engine=INNODB;

create table REFLEX(

```

```

        ID_Macchina int not null,
        MotoreAF bit not null,
        primary key (ID_Macchina),
        foreign key (ID_Macchina) references MACCHINA (ID)
    ) engine=INNODB;

create table MONTA(
    ID_Macchina int not null,
    ID_Obiettivo int not null,
    Attacco varchar(10) not null,
    primary key (ID_Macchina, ID_Obiettivo),
    foreign key (ID_Macchina) references MACCHINA (ID),
    foreign key (ID_Obiettivo) references OBIETTIVO (ID)
) engine=INNODB;

create table FOTOGRAFIA(
    ID int not null AUTO_INCREMENT,
    Descrizione varchar(140) not null,
    Data date,
    ID_Album int not null,
    ID_Categoria int not null,
    ID_Luogo int not null,
    ID_Macchina int not null,
    ID_Obiettivo int not null,
    LunghezzaFocale float not null,
    Angolo int not null,
    Diaframma float not null,
    Esposizione float not null,
    ISO int not null,
    Risoluzione int not null,
    primary key (ID),
    foreign key (ID_Album) references ALBUM (ID),
    foreign key (ID_Categoria) references CATEGORIA (ID),
    foreign key (ID_Luogo) references LUOGO (ID),
    foreign key (ID_Macchina) references MACCHINA (ID),
    foreign key (ID_Obiettivo) references OBIETTIVO (ID)
) engine=INNODB;

create table VOTO(
    ID int not null AUTO_INCREMENT,
    ID_Utente int not null,
    ID_Foto int not null,
    Voto int not null,
    Descrizione varchar(140) not null,
    primary key (ID),
    foreign key (ID_Utente) references UTENTE (ID),
    foreign key (ID_Foto) references FOTOGRAFIA (ID)
) engine=INNODB;

```

4 - Inserimenti di Prova

Provincia

```
insert into PROVINCIA (Nome,Sigla) value ("Viterbo", "VT");
insert into PROVINCIA (Nome,Sigla) value ("Roma", "Rm");
insert into PROVINCIA (Nome,Sigla) value ("Milano", "MI");
```

ID	Nome	Sigla
1	Viterbo	VT
2	Roma	Rm
3	Milano	MI

3 rows in set (0,00 sec)

Citta

```
insert into CITTA (Nome, ID_Provincia) value ("Montefiascone", 1);
insert into CITTA (Nome, ID_Provincia) value ("Viterbo", 1);
insert into CITTA (Nome, ID_Provincia) value ("Roma", 2);
insert into CITTA (Nome, ID_Provincia) value ("Tivoli", 2);
insert into CITTA (Nome, ID_Provincia) value ("Legnano", 3);
```

ID	Nome	ID_Provincia
1	Montefiascone	1
2	Viterbo	1
3	Roma	2
4	Tivoli	2
5	Legnano	3

Luogo

```
insert into LUOGO (Via, ID_Citta) value ("AB", 1);
insert into LUOGO (Via, ID_Citta) value ("CA", 2);
insert into LUOGO (Via, ID_Citta) value ("BB", 3);
insert into LUOGO (Via, ID_Citta) value ("AB", 4);
insert into LUOGO (Via, ID_Citta) value ("BB", 5);
```

ID	Via	ID_Citta
1	AB	1
2	CA	2
3	BB	3
4	AB	4
5	BB	5

Studio

```
insert into STUDIO (Nome) value ("StudioVT");
insert into STUDIO (Nome) value ("StudioRM1");
```

```
insert into STUDIO (Nome) value ("StudioRM2");
```

ID	Nome
1	StudioVT
2	StudioRM1
3	StudioRM2

Risiede

```
insert into RISIEDE (ID_Studio, ID_Luogo, NumeroCivico) value (1, 1, 21);
insert into RISIEDE (ID_Studio, ID_Luogo, NumeroCivico) value (1, 2, 3);
insert into RISIEDE (ID_Studio, ID_Luogo, NumeroCivico) value (2, 3, 4);
insert into RISIEDE (ID_Studio, ID_Luogo, NumeroCivico) value (3, 4, 65);
insert into RISIEDE (ID_Studio, ID_Luogo, NumeroCivico) value (2, 5, 121);
```

ID_Studio	ID_Luogo	NumeroCivico
1	1	21
1	2	3
2	3	4
2	5	121
3	4	65

Utente

```
insert into UTENTE (Nome, Cognome, Data, Email, Cellulare, ID_Studio) value
("Donato", "Stanco", '1998-06-26', "dstanco@hotmail.it", "3468028262", 1);
insert into UTENTE (Nome, Cognome, Data, Email, Cellulare, ID_Studio) value
("Marco", "Rossi", '1999-04-29', "marcoro99@gmail.com", "3353498763", 1);
insert into UTENTE (Nome, Cognome, Data, Email, Cellulare, ID_Studio) value
("Giacomo", "Verdi", '1992-11-12', "giacomove92@gmail.com", "3319303549", 2);
insert into UTENTE (Nome, Cognome, Data, Email, Cellulare, ID_Studio) value
("Matteo", "Gialli", '1993-09-02', "matteogi93@gmail.com", "3330382384", 2);
insert into UTENTE (Nome, Cognome, Data, Email, Cellulare, ID_Studio) value
("Luca", "Neri", '1996-02-24', "lucane96@gmail.com", "3339412345", 3);
```

ID	Nome	Cognome	Data	Email	Cellulare	ID_Studio
1	Donato	Stanco	1998-06-26	dstanco@hotmail.it	3468028262	1
2	Marco	Rossi	1999-04-29	marcoro99@gmail.com	3353498763	1
3	Giacomo	Verdi	1992-11-12	giacomove92@gmail.com	3319303549	2
4	Matteo	Gialli	1993-09-02	matteogi93@gmail.com	3330382384	2
5	Luca	Neri	1996-02-24	lucane96@gmail.com	3339412345	3

Album

```
insert into ALBUM (NomeAlbum, Data, ID_Utente) value ("Paesaggi", '2019-02-22', 1);
insert into ALBUM (NomeAlbum, Data, ID_Utente) value ("Animali", '2019-04-02', 1);
insert into ALBUM (NomeAlbum, Data, ID_Utente) value ("Ritratti", '2018-02-22', 2);
insert into ALBUM (NomeAlbum, Data, ID_Utente) value ("Citta", '2019-06-10', 2);
insert into ALBUM (NomeAlbum, Data, ID_Utente) value ("Ritratti", '2019-01-23', 3);
insert into ALBUM (NomeAlbum, Data, ID_Utente) value ("Edifici", '2019-11-01', 4);
insert into ALBUM (NomeAlbum, Data, ID_Utente) value ("Paesaggi", '2019-12-22', 5);
```

ID	NomeAlbum	Data	ID_Utente
1	Paesaggi	2019-02-22	1
2	Animali	2019-04-02	1
3	Ritratti	2018-02-22	2
4	Citta	2019-06-10	2
5	Ritratti	2019-01-23	3
6	Edifici	2019-11-01	4
7	Paesaggi	2019-12-22	5

Categoria

```
insert into CATEGORIA (Nome, Descrizione) value ("Natura", "abcd");
insert into CATEGORIA (Nome, Descrizione) value ("Persone", "efgh");
insert into CATEGORIA (Nome, Descrizione) value ("Oggetti", "ilmn");
```

ID	Nome	Descrizione
1	Natura	abcd
2	Persone	efgh
3	Oggetti	ilmn

Marca

```
insert into MARCA (Nome, Stato) value ("Nikon", "Giappone");
insert into MARCA (Nome, Stato) value ("Sony", "Giappone");
insert into MARCA (Nome, Stato) value ("Leica", "Germania");
```

ID	Nome	Stato
1	Nikon	Giappone
2	Sony	Giappone
3	Leica	Germania

Macchina

```
insert into MACCHINA (Modello, Sensore, Peso, MaxISO, ID_Marca) value ("ni12", 43, 280, 6400, 1);
insert into MACCHINA (Modello, Sensore, Peso, MaxISO, ID_Marca) value ("ni34", 28, 230, 5000, 1);
insert into MACCHINA (Modello, Sensore, Peso, MaxISO, ID_Marca) value ("sn07", 43, 300, 4000, 2);
insert into MACCHINA (Modello, Sensore, Peso, MaxISO, ID_Marca) value ("sn12", 21, 200, 5000, 2);
insert into MACCHINA (Modello, Sensore, Peso, MaxISO, ID_Marca) value ("Le32", 54, 400, 6400, 3);
```

ID	Modello	Sensore	Peso	MaxISO	ID_Marca
1	ni12	43	280	6400	1
2	ni34	28	230	5000	1
3	sn07	43	300	4000	2
4	sn12	21	200	5000	2
5	Le32	54	400	6400	3

Obiettivo

```

insert into OBIETTIVO (Modello, OIS, AF, Zoom, ID_Marca) value ("Obiettivo1", 0,
1, 50, 1);
insert into OBIETTIVO (Modello, OIS, AF, Zoom, ID_Marca) value ("Obiettivo2", 1,
1, 100, 2);
insert into OBIETTIVO (Modello, OIS, AF, Zoom, ID_Marca) value ("Obiettivo3", 1,
0, 100, 3);
insert into OBIETTIVO (Modello, OIS, AF, Zoom, ID_Marca) value ("Obiettivo4", 0,
1, 50, 2);
insert into OBIETTIVO (Modello, OIS, AF, Zoom, ID_Marca) value ("Obiettivo5", 1,
1, 100, 1);

```

ID	Modello	OIS	AF	Zoom	ID_Marca
1	Obiettivo1		0	50	1
2	Obiettivo2	1	1	100	2
3	Obiettivo3	1		100	3
4	Obiettivo4		1	50	2
5	Obiettivo5	1	1	100	1

Compatta

```

insert into COMPATTA (ID_Macchina, ID_Obiettivo, ZoomDigitale,
StabilizzazioneDigitale) value (3, 2, 100, 1);
insert into COMPATTA (ID_Macchina, ID_Obiettivo, ZoomDigitale,
StabilizzazioneDigitale) value (5, 3, 100, 0);

```

ID_Macchina	ID_Obiettivo	ZoomDigitale	StabilizzazioneDigitale
3	2	100	1
5	3	100	0

Reflex

```

insert into REFLEX (ID_Macchina, MotoreAF) value (1, 0);
insert into REFLEX (ID_Macchina, MotoreAF) value (2, 1);
insert into REFLEX (ID_Macchina, MotoreAF) value (4, 1);

```

ID_Macchina	MotoreAF
1	
2	2
4	2

Monta

```
insert into MONTA (ID_Macchina, ID_Obiettivo, Attacco) value (1, 1, "BF");
insert into MONTA (ID_Macchina, ID_Obiettivo, Attacco) value (1, 5, "BF");
insert into MONTA (ID_Macchina, ID_Obiettivo, Attacco) value (2, 1, "BF");
insert into MONTA (ID_Macchina, ID_Obiettivo, Attacco) value (2, 5, "BF");
insert into MONTA (ID_Macchina, ID_Obiettivo, Attacco) value (4, 2, "CS");
```

ID_Macchina	ID_Obiettivo	Attacco
1	1	BF
1	5	BF
2	1	BF
2	5	BF
4	2	CS

Fotografia

```
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("a", '2019-01-01', 1, 1, 1, 1, 5, 52, 70, 2.0, 10, 200,
1080);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("b", '2019-03-02', 2, 1, 2, 1, 1, 44, 110, 2.5, 80, 1000,
1440);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("c", '2019-02-03', 2, 2, 3, 3, 4, 48, 24, 2.3, 80, 1200,
720);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("d", '2019-06-04', 3, 3, 1, 2, 1, 44, 70, 2.2, 40, 1300,
1440);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("e", '2019-06-04', 4, 3, 4, 2, 5, 52, 120, 2.1, 30, 800,
1080);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("f", '2019-04-05', 4, 3, 5, 2, 5, 52, 16, 1.8, 20, 200,
1440);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("g", '2019-01-06', 5, 1, 1, 4, 2, 72, 70, 1.7, 40, 100,
1080);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
```



```

Risoluzione) value ("h", '2019-01-06', 5, 2, 2, 4, 2, 72, 70, 1.4, 60, 300,
720);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("i", '2019-08-07', 6, 2, 5, 5, 3, 52, 120, 1.9, 90, 700,
1440);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("l", '2019-08-07', 6, 2, 3, 5, 3, 52, 180, 1.8, 10, 1200,
720);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("m", '2019-11-08', 7, 2, 1, 3, 4, 48, 44, 2.2, 60, 2200,
1440);
insert into FOTOGRAFIA (Descrizione, Data, ID_Album, ID_Categoria, ID_Luogo,
ID_Macchina, ID_Obiettivo, LunghezzaFocale, Angolo, Diaframma, Esposizione, ISO,
Risoluzione) value ("n", '2019-12-09', 7, 1, 3, 2, 5, 52, 70, 2.0, 30, 2600,
1440);

```

ID	Descrizione	Data	ID_Album	ID_Categoria	ID_Luogo	ID_Macchina	ID_Obiettivo	LunghezzaFocale	Angolo	Diaframma	Esposizione	ISO	Risoluzione
1	a	2019-01-01	1	1	1	1	5	52	70	2	10	200	1080
2	b	2019-03-02	2	1	2	1	1	44	110	2.5	80	1000	1440
3	c	2019-02-03	2	2	3	3	4	48	24	2.3	80	1200	720
4	d	2019-06-04	3	3	1	2	1	44	70	2.2	40	1300	1440
5	e	2019-06-04	4	3	4	2	5	52	120	2.1	30	800	1080
6	f	2019-04-05	4	3	5	2	5	52	16	1.8	20	200	1440
7	g	2019-01-06	5	1	1	4	2	72	70	1.7	40	100	1080
8	h	2019-01-06	5	2	2	4	2	72	70	1.4	60	300	720
9	i	2019-08-07	6	2	5	5	3	52	120	1.9	90	700	1440
10	l	2019-08-07	6	2	3	5	3	52	180	1.8	10	1200	720
11	m	2019-11-08	7	2	1	3	4	48	44	2.2	60	2200	1440
12	n	2019-12-09	7	1	3	2	5	52	70	2	30	2600	1440

Voto

```

insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (1, 1, 8,
"abcdefg");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (1, 5, 6,
"jfhldsdg");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (1, 6, 8,
"abcdefg");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (1, 7, 7,
"jfhldsdg");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (1, 8, 5,
"abcdefg");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (1, 9, 6,
"jfhldsdg");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (2, 10, 7,
"gfghfhsfhf");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (2, 11, 5,
"dfhfdffd");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (2, 10, 10,
"dfhfdffd");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (2, 8, 7,
"gfghfhsfhf");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (3, 1, 8,
"fdfdhfhf");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (3, 2, 7,
"dfhhghsfreh");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (3, 3, 8,
"fdfdhfhf");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (3, 6, 7,
"dfhhghsfreh");

```



```

insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (4, 7, 6,
"bnbnhytery");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (4, 8, 5,
"cuyowcieui");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (4, 1, 6,
"bnbnhytery");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (4, 2, 8,
"cuyowcieui");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (5, 1, 9,
"dey7398hce");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (5, 5, 7,
"cbuebcuoee");
insert into VOTO (ID_Utente, ID_Foto, Voto, Descrizione) value (5, 6, 4,
"equuicoie");

```

ID	ID_Utente	ID_Foto	Voto	Descrizione
1	1	1	8	abcdefg
2	1	5	6	jfhldsdg
3	1	6	8	abcdefg
4	1	7	7	jfhldsdg
5	1	8	5	abcdefg
6	1	9	6	jfhldsdg
7	2	10	7	gfhgfhshfh
8	2	11	5	dfhfdffd
9	2	10	10	dfhfdffd
10	2	8	7	gfhgfhshfh
11	3	1	8	fdfdhfhf
12	3	2	7	dfhhghsfreh
13	3	3	8	fdfdhfhf
14	3	6	7	dfhhghsfreh
15	4	7	6	bnbnhytery
16	4	8	5	cuyowcieui
17	4	1	6	bnbnhytery
18	4	2	8	cuyowcieui
19	5	1	9	dey7398hce
20	5	5	7	cbuebcuoee
21	5	6	4	equuicoie

5 - Query

1-Mostrare il numero di foto per ogni categoria

```
SELECT Categoria.Nome, COUNT(*) as NumeroFoto
FROM FOTOGRAFIA, CATEGORIA
WHERE FOTOGRAFIA.ID_Categoria = CATEGORIA.ID
GROUP BY CATEGORIA.Nome;
```

Nome	NumeroFoto
Natura	4
Persone	5
Oggetti	3

2-Mostrare il numero di utenti registrati raggruppandoli per anno di nascita

```
SELECT YEAR(UTENTE.Data) as Anno, COUNT(UTENTE.ID) AS NumeroUtenti
FROM UTENTE
GROUP BY YEAR(UTENTE.Data);
```

Anno	NumeroUtenti
1998	1
1999	1
1992	1
1993	1
1996	1

3-Mostrare gli utenti nati dopo il 1996

```
SELECT UTENTE.ID, UTENTE.Nome, UTENTE.Cognome, UTENTE.Data
FROM UTENTE
WHERE YEAR(UTENTE.Data) >= 1995;
```

ID	Nome	Cognome	Data
1	Donato	Stanco	1998-06-26
2	Marco	Rossi	1999-04-29
5	Luca	Neri	1996-02-24

4-Mostrare gli utenti che lavorano per gli studi che hanno almeno una sede nella provincia di Roma

```
SELECT UTENTE.ID, UTENTE.Nome, UTENTE.Cognome
FROM UTENTE, STUDIO, RISIEDE, LUOGO, CITTA, PROVINCIA
WHERE UTENTE.ID_Studio = STUDIO.ID and RISIEDE.ID_Studio = STUDIO.ID and
RISIEDE.ID_Luogo = LUOGO.ID and LUOGO.ID_Citta = CITTA.ID and CITTA.ID_Provincia
= PROVINCIA.ID and PROVINCIA.Nome = "Roma";
```

ID	Nome	Cognome
3	Giacomo	Verdi
4	Matteo	Gialli
5	Luca	Neri

5-Mostrare il numero di studi presenti nella provincia di Roma

```
SELECT COUNT(*) as NumeroStudiRoma
FROM RISIEDE, LUOGO, CITTA, PROVINCIA
WHERE RISIEDE.ID_Luogo = LUOGO.ID and LUOGO.ID_Citta = CITTA.ID and
CITTA.ID_Provincia = PROVINCIA.ID and PROVINCIA.Nome = "Roma";
```

NumeroStudiRoma
2

6-Mostrare il numero di studi presenti in ciascuna provincia

```
SELECT PROVINCIA.Nome, COUNT(*) as NumeroStudi
FROM RISIEDE, LUOGO, CITTA, PROVINCIA
WHERE RISIEDE.ID_Luogo = LUOGO.ID and LUOGO.ID_Citta = CITTA.ID and
CITTA.ID_Provincia = PROVINCIA.ID
GROUP BY PROVINCIA.ID;
```

Nome	NumeroStudi
Viterbo	2
Roma	2
Milano	1

7-Mostrare il numero di album pubblicati da ogni utente in ordine decrescente

```
SELECT UTENTE.ID, UTENTE.Nome, UTENTE.Cognome, COUNT(ALBUM.ID) as NumeroAlbum
FROM ALBUM, UTENTE
WHERE ALBUM.ID_Utente = UTENTE.ID
GROUP BY ALBUM.ID_Utente
ORDER BY COUNT(ALBUM.ID) DESC;
```

ID	Nome	Cognome	NumeroAlbum
1	Donato	Stanco	2
2	Marco	Rossi	2
3	Giacomo	Verdi	1
4	Matteo	Gialli	1
5	Luca	Neri	1

8-Mostare il numero di fotografie scattate da ciascuna macchina con il TOTALE finale

```
SELECT COALESCE(FOTOGRAFIA.ID_Macchina, 'TOTALE') as ID_Macchina, MARCA.Nome as
Marca, MACCHINA.Modello as Modello, COUNT(*) as NumFoto
FROM FOTOGRAFIA, MACCHINA, MARCA
WHERE FOTOGRAFIA.ID_Macchina = MACCHINA.ID and MACCHINA.ID_Marca = MARCA.ID
GROUP BY FOTOGRAFIA.ID_Macchina WITH ROLLUP;
```

ID_Macchina	Marca	Modello	NumFoto
1	Nikon	ni12	2
2	Nikon	ni34	4
3	Sony	sn07	2
4	Sony	sn12	2
5	Leica	Le32	2
TOTALE	Leica	Le32	12

9-Stampare gli utenti che hanno il numero di cellulare che inizia per 346

```
SELECT UTENTE.ID, UTENTE.Nome, UTENTE.Cognome, UTENTE.Cellulare
FROM UTENTE
WHERE UTENTE.Cellulare like "346%";
```

ID	Nome	Cognome	Cellulare
1	Donato	Stanco	3468028262

10-Per ogni utente fare una media dei voti delle proprie foto

```
SELECT UTENTE.ID, UTENTE.Nome, UTENTE.Cognome, AVG(VOTO.Voto) as MediaVoti
FROM UTENTE, VOTO
WHERE VOTO.ID_Utente = UTENTE.ID
GROUP BY UTENTE.ID
ORDER BY AVG(VOTO.Voto);
```

ID	Nome	Cognome	MediaVoti
4	Matteo	Gialli	6.2500
1	Donato	Stanco	6.6667
5	Luca	Neri	6.6667
2	Marco	Rossi	7.2500
3	Giacomo	Verdi	7.5000

11-Stampare l'album con la media voti minore

```
SELECT ALBUM.ID as ID_Album, ALBUM.NomeAlbum, UTENTE.ID as ID_Utente,
UTENTE.Nome, UTENTE.Cognome, AVG(VOTO.Voto) as MediaVoti
FROM UTENTE, ALBUM, VOTO, FOTOGRAFIA
WHERE VOTO.ID_Foto = FOTOGRAFIA.ID and FOTOGRAFIA.ID_Album = ALBUM.ID and
ALBUM.ID_Utente = UTENTE.ID
GROUP BY FOTOGRAFIA.ID_Album
ORDER BY AVG(VOTO.Voto) LIMIT 1;
```

ID_Album	NomeAlbum	ID_Utente	Nome	Cognome	MediaVoti
7	Paesaggi	5	Luca	Neri	5.0000

12-Stampare l'album con più voti

```
SELECT ALBUM.ID as ID_Album, ALBUM.NomeAlbum, UTENTE.ID as ID_Utente,
UTENTE.Nome, UTENTE.Cognome, COUNT(VOTO.Voto) as MediaVoti
FROM UTENTE, ALBUM, VOTO, FOTOGRAFIA
WHERE VOTO.ID_Foto = FOTOGRAFIA.ID and FOTOGRAFIA.ID_Album = ALBUM.ID and
ALBUM.ID_Utente = UTENTE.ID
GROUP BY FOTOGRAFIA.ID_Album
ORDER BY COUNT(*) DESC LIMIT 1;
```

ID_Album	NomeAlbum	ID_Utente	Nome	Cognome	MediaVoti
4	Citta	2	Marco	Rossi	5

13-Stampare il modello di macchina più usato

```
SELECT FOTOGRAFIA.ID_Macchina, MARCA.Nome as Marca, MACCHINA.Modello as Modello,
COUNT(*) as NumFoto
FROM FOTOGRAFIA, MACCHINA, MARCA
WHERE FOTOGRAFIA.ID_Macchina = MACCHINA.ID and Macchina.ID_Marca = MARCA.ID
GROUP BY FOTOGRAFIA.ID_Macchina
ORDER BY COUNT(*) DESC LIMIT 1;
```

ID_Macchina	Marca	Modello	NumFoto
2	Nikon	ni34	4

14-Stampare le macchine compatibili solo con l'Obiettivo5

```
SELECT MONTA.Attacco, MACCHINA.Modello, MARCA.nome as Marca
FROM MONTA, MACCHINA, MARCA, OBIETTIVO
WHERE MACCHINA.ID_Marca = MARCA.ID and MONTA.ID_Macchina = MACCHINA.ID and
MONTA.ID_Obiettivo = OBIETTIVO.ID and OBIETTIVO.Modello = "Obiettivo5";
```

Attacco	Modello	Marca
BF	ni12	Nikon
BF	ni34	Nikon

15-Trovare gli utenti che hanno pubblicato un album dopo il giorno in cui l'utente con ID=1 ha pubblicato l'album Animali

```
SELECT UTENTE.ID as ID_Utente, UTENTE.Nome, UTENTE.Cognome, ALBUM.ID as
ID_Album, ALBUM.NomeAlbum, ALBUM.Data
FROM UTENTE, ALBUM, ALBUM as ALB
```

WHERE ALBUM.ID_Utente = UTENTE.ID and ALBUM.Data > ALB.Data and ALB.NomeAlbum = "Animali" and ALB.ID_Utente = 1;

ID_Utente	Nome	Cognome	ID_Album	NomeAlbum	Data
2	Marco	Rossi	4	Citta	2019-06-10
4	Matteo	Gialli	6	Edifici	2019-11-01
5	Luca	Neri	7	Paesaggi	2019-12-22

16-Trovare gli utenti che hanno messo lo stesso voto dell'utente 1 alla foto con ID=7

SELECT UTENTE.ID as ID_Utente, UTENTE.Nome, UTENTE.Cognome, VOTO.Voto
FROM UTENTE, VOTO, VOTO as VT
WHERE VOTO.ID_Utente = UTENTE.ID and VOTO.ID_Foto = VT.ID_Foto and VOTO.Voto = VT.Voto and VT.ID_Utente = 1 and VT.ID_Foto = 8;

ID_Utente	Nome	Cognome	Voto
1	Donato	Stanco	5
4	Matteo	Gialli	5

17-Mostrare tutte le foto, e da chi sono state scattate, che hanno un angolo superiore alla foto con ID=2

SELECT UTENTE.ID as ID_Utente, UTENTE.Nome, UTENTE.Cognome, FOTOGRAFIA.ID as ID_Foto, ALBUM.NomeAlbum, FOTOGRAFIA.Angolo
FROM UTENTE, ALBUM, FOTOGRAFIA, FOTOGRAFIA as FO
WHERE ALBUM.ID_Utente = UTENTE.ID and FOTOGRAFIA.ID_Album = ALBUM.ID and FOTOGRAFIA.Angolo > FO.Angolo and FO.ID=2;

ID_Utente	Nome	Cognome	ID_Foto	NomeAlbum	Angolo
2	Marco	Rossi	5	Citta	120
4	Matteo	Gialli	9	Edifici	120
4	Matteo	Gialli	10	Edifici	180

18-Mostrare tutte le foto pubblicate tra il 2019-06-01 e il 2019-09-01 e da chi

SELECT UTENTE.Nome, UTENTE.Cognome, ALBUM.NomeAlbum, FOTOGRAFIA.ID as ID_Foto, FOTOGRAFIA.Data
FROM FOTOGRAFIA, ALBUM, UTENTE
WHERE ALBUM.ID_Utente = UTENTE.ID and FOTOGRAFIA.ID_Album = ALBUM.ID and FOTOGRAFIA.Data between '2019-06-01' and '2019-09-01';

Nome	Cognome	NomeAlbum	ID_Foto	Data
Marco	Rossi	Ritratti	4	2019-06-04
Marco	Rossi	Citta	5	2019-06-04
Matteo	Gialli	Edifici	9	2019-08-07
Matteo	Gialli	Edifici	10	2019-08-07

19-Mostrare tutte le foto pubblicate nel mese di Giugno di qualsiasi anno

```
SELECT UTENTE.ID as ID_Utente, UTENTE.Nome, UTENTE.Cognome, ALBUM.NomeAlbum,
FOTOGRAFIA.ID as ID_Foto, FOTOGRAFIA.data
FROM FOTOGRAFIA, ALBUM, UTENTE
WHERE FOTOGRAFIA.ID_Album = ALBUM.ID and ALBUM.ID_Utente = UTENTE.ID and
MONTH(FOTOGRAFIA.Data) = 6;
```

ID_Utente	Nome	Cognome	NomeAlbum	ID_Foto	data
2	Marco	Rossi	Ritratti	4	2019-06-04
2	Marco	Rossi	Citta	5	2019-06-04

20-Stampare tutti gli utenti che hanno scattato foto nelle stesse Citta in cui ha scattato l'utente Donato Stanco

```
SELECT DISTINCT UTENTE1.ID as ID_Utente, UTENTE1.Nome, UTENTE1.Cognome,
CITTA.Nome
FROM UTENTE, ALBUM, FOTOGRAFIA, LUOGO, CITTA, UTENTE as UTENTE1, ALBUM as
ALBUM1, FOTOGRAFIA as FOTOGRAFIA1, LUOGO as LUOGO1, CITTA as CITTA1
WHERE ALBUM1.ID_Utente = UTENTE1.ID and FOTOGRAFIA1.ID_Album = ALBUM1.ID and
FOTOGRAFIA1.ID_Luogo = LUOGO1.ID and
LUOGO1.ID_Citta = CITTA.ID and ALBUM.ID_Utente = UTENTE.ID and
FOTOGRAFIA.ID_Album = ALBUM.ID and FOTOGRAFIA.ID_Luogo = LUOGO.ID and
LUOGO.ID_Citta = CITTA.ID and UTENTE.Nome = "Donato" and
UTENTE.Cognome = "Stanco";
```

ID_Utente	Nome	Cognome	Nome
1	Donato	Stanco	Montefiascone
2	Marco	Rossi	Montefiascone
3	Giacomo	Verdi	Montefiascone
5	Luca	Neri	Montefiascone
1	Donato	Stanco	Viterbo
3	Giacomo	Verdi	Viterbo
1	Donato	Stanco	Roma
4	Matteo	Gialli	Roma
5	Luca	Neri	Roma

21-Stampare l'obiettivo usato più volte insieme alla macchina ni34

```
SELECT FOTOGRAFIA.ID_Obiettivo, OBIETTIVO.Modello as ModelloObiettivo,
COUNT(FOTOGRAFIA.ID_Obiettivo) as UsatoNUM, MARCA.Nome as Marca
FROM FOTOGRAFIA, MACCHINA, OBIETTIVO, MARCA
WHERE OBIETTIVO.ID_Marca = MARCA.ID and FOTOGRAFIA.ID_Obiettivo = OBIETTIVO.ID
and FOTOGRAFIA.ID_Macchina = MACCHINA.ID and MACCHINA.Modello = "ni34"
GROUP BY FOTOGRAFIA.ID_Obiettivo
ORDER BY COUNT(FOTOGRAFIA.ID_Obiettivo) DESC LIMIT 1;
```

ID_Obiettivo	ModelloObiettivo	UsatoNUM	Marca
5	Obiettivo5	3	Nikon

22-Stampare la macchina più usata nella provincia di Roma

```
SELECT FOTOGRAFIA.ID_Macchina, MACCHINA.Modello as ModelloMacchina, MARCA.Nome
as NomeMarca, COUNT(FOTOGRAFIA.ID_Macchina) as UsatoNum
FROM FOTOGRAFIA, MACCHINA, MARCA, LUOGO, CITTA, PROVINCIA
WHERE MACCHINA.ID_Marca = MARCA.ID and FOTOGRAFIA.ID_Macchina = MACCHINA.ID and
FOTOGRAFIA.ID_Luogo = LUOGO.ID and LUOGO.ID_Citta = CITTA.ID and
CITTA.ID_Provincia = PROVINCIA.ID and PROVINCIA.Nome = "Roma"
GROUP BY FOTOGRAFIA.ID_Macchina
ORDER BY COUNT(FOTOGRAFIA.ID_Macchina) DESC LIMIT 1;
```

ID_Macchina	ModelloMacchina	NomeMarca	UsatoNum
2	ni34	Nikon	2

23-Visualizzare tutti i commenti di una determinata foto, (ad esempio quella con ID=2)

```
SELECT VOTO.Voto, VOTO.Descrizione, UTENTE.Nome, UTENTE.Cognome
FROM VOTO, UTENTE
WHERE VOTO.ID_Utente = UTENTE.ID and VOTO.ID_Foto = 2;
```

Voto	Descrizione	Nome	Cognome
7	dfhhghsfreh	Giacomo	Verdi
8	cuyowcieui	Matteo	Gialli

24-Visualizzare quanti modelli di macchine produce ogni paese

```
SELECT MARCA.Stato, COUNT(*) as NumModelli
FROM MARCA, MACCHINA
WHERE MACCHINA.ID_Marca = MARCA.ID
GROUP BY MARCA.Stato;
```

Stato	NumModelli
Giappone	4
Germania	1

6 - Algebra e Calcolo Relazionale

Traduzione in algebra relazionale della query numero 17:

Mostrare tutte le foto (e da chi sono scattate) che hanno angolo superiore alla foto con ID=2

```

$$\Pi_{[UTENTE.ID, UTENTE.Nome, UTENTE.Cognome, FOTOGRAFIA.ID, ALBUM.NomeAlbum, FOTOGRAFIA.Angolo]} \left( \right. \\ \quad UTENTE \mid X \mid [ID = ID\_Utente] \left( ALBUM \mid X \mid [ID = ID\_Album] \left( \right. \right. \\ \quad \quad \sigma_{[F.Angolo > FOTOGRAFIA.Angolo]} \left( \rho_{[F \leftarrow FOTOGRAFIA]} \left( \left( \sigma_{[FOTOGRAFIA.ID = 2]} FOTOGRAFIA \right) \right) \right) \\ \left. \right) \left. \right)$$

```


Traduzione in calcolo relazionale della query numero 4:

Mostrare gli utenti che lavorano per gli studi che hanno almeno una sede nella provincia di Roma

```
{u.ID, u.Nome, u.Cognome |  
  u(UTENTE), s(STUDIO), r(RISIEDE), l(LUOGO), c(CITTA), p(PROVINCIA) |  
  u.ID_Studio = s.ID ∧ r.ID_Studio = s.ID ∧ r.ID_Luogo = l.ID ∧  
  l.ID_Citta = c.ID ∧ c.ID_Provincia = p.ID ∧ p.Nome = "Roma"}
```

7 - Parti Ulteriori del DB

7.1 - TRIGGER

1-Trigger che verifica il corretto formato della mail dell'Utente

```
DELIMITER //  
CREATE TRIGGER TR_UTENTE_Email  
  BEFORE INSERT ON UTENTE  
  FOR EACH ROW  
  BEGIN  
    IF NEW.Email not like "%@gmail.com" OR "%@outlook.com" OR  
      "%@yahoo.com" OR "%@hotmail.com" THEN SIGNAL SQLSTATE '45000'  
    SET MESSAGE_TEXT = 'Dominio Email non riconosciuto';  
    END IF;  
  END; //  
DELIMITER ;  
DROP TRIGGER TR_UTENTE_Email;
```

2-Trigger che verifica che il numero di cellulare dell'Utente sia valido

```
DELIMITER //  
CREATE TRIGGER TR_UTENTE_Cellulare  
  BEFORE INSERT ON UTENTE  
  FOR EACH ROW  
  BEGIN  
    IF LENGTH(NEW.Cellulare) != 10 THEN SIGNAL SQLSTATE '45000'  
    SET MESSAGE_TEXT = 'Numero di cellulare non valido (lunghezza non valida)';  
    ELSEIF NEW.Cellulare NOT REGEXP '^[0-9]+$' THEN SIGNAL SQLSTATE '45000'  
    SET MESSAGE_TEXT = 'Numero di cellulare non valido (sono presenti  
caratteri)';  
    END IF;  
  END; //  
DELIMITER ;  
DROP TRIGGER TR_UTENTE_Cellulare;
```

3-Trigger che verifica che l'utente sia maggiorenne

```
DELIMITER //  
CREATE TRIGGER TR_UTENTE_Data  
  BEFORE INSERT ON UTENTE  
  FOR EACH ROW  
  BEGIN  
    IF ADDDATE(NEW.Data, INTERVAL 18 YEAR) >= CURRENT_DATE() THEN SIGNAL  
SQLSTATE '45000'  
    SET MESSAGE_TEXT = 'Utente ancora non maggiorenne';  
    END IF;  
  END; //  
DELIMITER ;  
DROP TRIGGER TR_UTENTE_Data;
```

4-Trigger che verifica la correttezza del voto inserito

```
DELIMITER //
CREATE TRIGGER TR_VOTO_Voto
  BEFORE INSERT ON VOTO
  FOR EACH ROW
  BEGIN
    IF NEW.Voto > 10 THEN
      SET NEW.Voto = 10;
    ELSEIF NEW.Voto < 0 THEN
      SET NEW.Voto = 0;
    END IF;
  END; //
DELIMITER ;
DROP TRIGGER TR_VOTO_Voto;
```

5-Trigger che verifica la correttezza della data inserita per l'album pubblicato dall'utente

```
DELIMITER //
CREATE TRIGGER TR_ALBUM_Data
  BEFORE INSERT ON ALBUM
  FOR EACH ROW
  BEGIN
    IF NEW.Data > CURRENT_DATE() THEN SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Data inserita errata';
    END IF;
  END; //
DELIMITER ;
DROP TRIGGER TR_ALBUM_Data;
```

6-Verifica la correttezza della data inserita per la foto pubblicata dall'utente

```
DELIMITER //
CREATE TRIGGER TR_FOTO_Data
  BEFORE INSERT ON FOTOGRAFIA
  FOR EACH ROW
  BEGIN
    IF NEW.Data > CURRENT_DATE() THEN SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Data inserita errata';
    END IF;
  END; //
DELIMITER ;
DROP TRIGGER TR_FOTO_Data;
```

7-Trigger che verifica la correttezza dell'angolo della foto scatta dall'utente

```
DELIMITER //
CREATE TRIGGER TR_FOTO_Angolo
  BEFORE INSERT ON FOTOGRAFIA
  FOR EACH ROW
  BEGIN
    IF NEW.Angolo > 220 THEN
      SET NEW.Voto = 220;
    ELSEIF NEW.Voto < 2 THEN
      SET NEW.Voto = 2;
    END IF;
  END; //
DELIMITER ;
DROP TRIGGER TR_FOTO_Angolo;
```

7.2 - Stored Procedure

1-Stored procedure che data una città restituisce tutte le foto scattate lì

```
DROP PROCEDURE IF NOT EXISTS FotoCitta;
DELIMITER //
CREATE PROCEDURE FotoCitta(IN citta VARCHAR(255))
BEGIN
    SELECT FOTOGRAFIA.* FROM FOTOGRAFIA, LUOGO, CITTA WHERE
    FOTOGRAFIA.ID_Luogo = LUOGO.ID AND LUOGO.ID_Citta = CITTA.ID AND CITTA.Nome =
    citta;
END //
DELIMITER ;
CALL FotoCitta ("Viterbo");
```

2-Stored procedure che calcola l'angolo della lente una volta inseriti la dimensione del sensore e la lunghezza focale della lente

```
DROP PROCEDURE IF NOT EXISTS proceduraAngolo;
DELIMITER //
CREATE PROCEDURE proceduraAngolo(IN LunghezzaFocale float, IN DimensioneSensore
float, OUT Angolo float)
BEGIN
    SET Angolo = DEGREES(2*ATAN((DimensioneSensore)/
(2*LunghezzaFocale)));
END //
DELIMITER ;
CALL proceduraAngolo(32, 43, @angolo);
SELECT @angolo;
```

3-Stored procedure che dati in input nome e cognome restituisce l'ID della prima persona corrispondente

```
DROP PROCEDURE IF EXISTS trovaUtente;
DELIMITER //
CREATE PROCEDURE trovaUtente(IN nome VARCHAR(20), cognome VARCHAR(20), OUT id
VARCHAR(20))
BEGIN
    DECLARE appoggio VARCHAR(20);
    DECLARE cercaUtente CURSOR FOR SELECT UTENTE.ID FROM UTENTE WHERE UTENTE.Nome
= nome and UTENTE.Cognome = cognome;
    OPEN cercaUtente;
    FETCH cercaUtente INTO appoggio;
    CLOSE cercaUtente;
    SET id = appoggio;
END; //
DELIMITER ;
CALL trovaUtente("Donato", "Stanco", @idUtente);
SELECT @idUtente;
```

7.3 - VIEW

1-Vista per calcolare la media dei voti per ogni album

```
CREATE VIEW MediaVotiAlbum AS (Media_NomeAlbum, Media_NomeUtente,
Media_MediaVoti)
SELECT ALBUM.ID as ID_Album, ALBUM.NomeAlbum, UTENTE.ID as ID_Utente,
UTENTE.Nome, UTENTE.Cognome, AVG(VOTO.Voto) as MediaVoti
FROM VOTO, FOTOGRAFIA, ALBUM, UTENTE
WHERE VOTO.ID_Foto = FOTOGRAFIA.ID and FOTOGRAFIA.ID_Album = ALBUM.ID and
    ALBUM.ID_Utente = UTENTE.ID
```

```
GROUP BY FOTOGRAFIA.ID_Album;
```

2-Vista per calcolare la media dei voti per ogni utente

```
CREATE VIEW MediaVotiAlbum AS (Media_NomeUtente, Media_MediaVoti)
SELECT UTENTE.ID as ID_Utente, UTENTE.Nome, UTENTE.Cognome, AVG(VOTO.Voto) as
MediaVoti
FROM VOTO, FOTOGRAFIA, ALBUM, UTENTE
WHERE ALBUM.ID_Utente = UTENTE.ID and FOTOGRAFIA.ID_Album = ALBUM.ID and
      VOTO.ID_Foto = FOTOGRAFIA.ID
GROUP BY UTENTE.ID;
```

8 - MONGO DB

8.1 - Collection

```
db.Provincia.insert([
    {"_id":0,
"Nome":"xRYKpLYLusIXtRWbHqNhGeMZfKtMewLxLeudSAy", "Sigla":"ml"},
    {"_id":1,
"Nome":"nJKSYwjchOfEQflTLbySFEoylwYVfJTSHLF", "Sigla":"SK"},
    {"_id":2,
"Nome":"MihtzuNkzfJwphnSjgJOWVZEHVpWKpdgwPMrTEEgcaAd", "Sigla":"WC"},
    {"_id":3, "Nome":"aZLqPwUjPGGPKSBbunlyogE",
"Sigla":"MI"}
])

db.Citta.insert([
    {"_id":0, "Nome":"oxWgWyuIDqvmjp", "Provincia":"1"},
    {"_id":1,
"Nome":"MSITnQsRxuJuRLnjoPzqGhOnnfGaVsDiRg", "Provincia":"1"},
    {"_id":2, "Nome":"rqjcCHUnckauPQdIIsfJ",
"Provincia":"2"},
    {"_id":3,
"Nome":"ybIgZzGoKPgPYffFMpkeCPWIjktmWkSAUjRnh", "Provincia":"3"},
    {"_id":4,
"Nome":"ubAUNgSsAMBSesRBfdTHGLUcxybrKmZcLqfjIggamlfSDMgNI", "Provincia":"3"},
])

db.Luogo.insert([
    {"_id":0,
"Via":"lOLypaqbPhSwXlqbPiRGWDqCfDcDhqrPQtvehioxwX", "Citta":"1"},
    {"_id":1, "Via":"upWffOG", "Citta":"1"},
    {"_id":2,
"Via":"LtnoecopObyEXucUgNCBBioXjwGnOMFqxQtMffv", "Citta":"2"},
    {"_id":3,
"Via":"BQsypakbUKRHBSatkBzkyIfUUukzPZhWLQyKQqtIBSLlh", "Citta":"2"},
    {"_id":4, "Via":"xKWTYHhXuwvxVXEePLYhZRM",
"Citta":"3"}
])

db.Studio.insert([
    {"_id":0, "Nome":"GvZVViazwchqpIGbqLWVlUSN"},
    {"_id":1,
"Nome":"IwMySOMmtQjOuhqNGLFpxrYJRJejpczVeWONVy"},
    {"_id":2, "Nome":"JNkGtLvlTMVmQJGYDQeE"},
    {"_id":3,
"Nome":"JObHpFOXqLjoGZVmfrOuqGhKHeInlNvwEvJGYWEL"},
    {"_id":4, "Nome":"vloITwxCOWSdnmoc"}
])
```

```

db.Risiede.insert([
    {"_id_Studio":1, "_id_Luogo":"1",
"NumeroCivico":141},
    {"_id_Studio":2, "_id_Luogo":"1",
"NumeroCivico":204},
    {"_id_Studio":0, "_id_Luogo":"2",
"NumeroCivico":46},
    {"_id_Studio":3, "_id_Luogo":"3",
"NumeroCivico":194},
    {"_id_Studio":4, "_id_Luogo":"4",
"NumeroCivico":202}
])

db.Utente.insert([
    {"_id":0, "Nome":"xXIpTiaAVZMksyOUerR",
"Cognome":"ZhNADmkaxYujGrnLhhGpQ", "Data":"2018-11-24",
"Email":"mKlZPXeZyOQsOdQCQJPvRbAxmZgyhLh", "Cellulare":"473981623",
"_id_Studio":"0"},
    {"_id":1,
"Nome":"IyroSglaNQzGanbwXWcodfIFAiSdtqfEquholj1XAGxznZG",
"Cognome":"xKGJxSQaPWbJvPZOcvuvBimkN", "Data":"2005-6-2",
"Email":"VgDIOSjMrEkgFpYqeZJPTpkEFXuvWMEmyLgpoDlOghMRXFITbR",
"Cellulare":"658131376", "_id_Studio":"0"},
    {"_id":2,
"Nome":"qJzPZKJlXZWhdVIVgHcAgLAWSyEschSMbxDYBl1meHtP",
"Cognome":"KUTJlgdIUdPzXZMpIfQVtHwgBZsEjnUSSVTqq", "Data":"2008-4-26",
"Email":"RsFWUfQpxIyjFiSOkdfdNTPwc", "Cellulare":"568183292", "_id_Studio":"1"},
    {"_id":3, "Nome":"mPoTrImA",
"Cognome":"RzAdJKrKJiMHdm", "Data":"1990-5-10", "Email":"ftRLawBuJztSZe",
"Cellulare":"165817483", "_id_Studio":"3"},
    {"_id":4, "Nome":"hAilMKzPKlBvDgYPMUzumkiwKTWogs",
"Cognome":"hyLubaXNsSBDUMYEGwPedfQIs", "Data":"2013-11-26",
"Email":"CyzjnTVNkSavEpdengYFXClfALAIeVukKCygYmmhNsz", "Cellulare":"686347699",
"_id_Studio":"1"}
])

db.Album.insert([
    {"_id":0, "NomeAlbum":"JBVqTneJie", "_id_Utente":1,
"Data":"2000-9-11"},
    {"_id":1,
"NomeAlbum":"rBOWtgYiTRLZFgtSiRxSDFtzeJqiDiGaww", "_id_Utente":2, "Data":"1986-
1-9"},
    {"_id":2,
"NomeAlbum":"dYhTnPeetltUutYxGRxsObgwLtbhXNKwLVonSvLVrGk", "_id_Utente":0,
"Data":"2008-1-2"},
    {"_id":3,
"NomeAlbum":"yzJgZmnvfIcJFVKfKcEZohubyiKVIjRYbfdiuMKbhme", "_id_Utente":3,
"Data":"1984-2-24"},
    {"_id":4,
"NomeAlbum":"ibOOkYNVGAWnUnQNMvzdZtZudzsVMMAiCJ", "_id_Utente":1, "Data":"1993-
1-14"}
])

db.Categoria.insert([
    {"_id":0, "Nome":"kHVLGeFSmrRpKdcoSzQFRcxqJACMGK",
"Descrizione":"uVLLSautsrwz"},
    {"_id":1, "Nome":"wzdMManMDqehwbANpsmjKr",
"Descrizione":"OWxohgIbDADrZVtpShtdcaQbpWpRoSJwQhHqFZwxJVted"},
    {"_id":2, "Nome":"KwYLMQYtk",
"Descrizione":"pBGaMFksQcYEknyR"},
    {"_id":3, "Nome":"iSlLijJTQbZESpbepFpAouxMqnkdZ",
"Descrizione":"NnJnaYiksbkKRQzbOShXBmzaPGdrUMKwXZdMejeuYrb"},

```

```

        {"_id":4, "Nome":"kkGekM",
"Descrizione":"PgoCOdZWUhREvAoRKNHMwkg"}
    ])

db.Marca.insert([
        {"_id":0, "Nome":"bkauLXwmXtktdOsELQJ", "Stato":
"nvQXMeWIS"},
        {"_id":1,
"Nome":"kHsqgBtjjSHHkFUEDjMWKvLPFhkzobhlssbHmUrEjPycD", "Stato":
"TtkbPsHCCEdAnVcTcrqooZYyPbmlPnoHHsAvrmNwIj"},
        {"_id":2, "Nome":"RjMULmDawROqdYBSjdrYkqsUCespwmVZ",
"Stato": "kVcKBJDYbdqmZsclMlJLIdVjj"},
        {"_id":3,
"Nome":"dCMEGeQAsDtDGBKZoiWbOHhltXVrdkqdfbmZDbZY", "Stato":
"lLPdoQgBVDSqecXjasJYHrmMDZiaqBQuvkdDAvToTwGuy"},
        {"_id":4,
"Nome":"janohqfOyqGkRAqeQfaZWDCqEbpGTskUIHCoP", "Stato":
"cwOwubvACTbzTZiTeGSqTobGPnPzODneZwxGTLdjBziSPC"}
    ])

db.Macchina.insert([
        {"_id":0, "Modello":"mAYRcqdgNTZwY",
"Sensore":93737, "Peso":36, "MaxISO":87, "_id_Marca":1},
        {"_id":1,
"Modello":"rMnAcMbJCvIVFlzMjGIAGNauKVFKEjYmlGVR", "Sensore":90534, "Peso":21,
"MaxISO":382, "_id_Marca":2},
        {"_id":2, "Modello":"hMpxnsxiZq", "Sensore":19840,
"Peso":62, "MaxISO":226, "_id_Marca":1},
        {"_id":3, "Modello":"EHjdFktExhxclIUcQDWhN",
"Sensore":89249, "Peso":18, "MaxISO":440, "_id_Marca":2},
        {"_id":4,
"Modello":"WRKJsVXFjvQlqdJtjpGyDTslqtrjHAhjQrpuhfPXYdt", "Sensore":37330,
"Peso":15, "MaxISO":194, "_id_Marca":3}
    ])

db.Obiettivo.insert([
        {"_id":0,
"Modello":"PIKcAiWNVAzwkfmJcwKFFTVBhYvmyMmTkqhBaMIGDhJxbRUfi", "OIS":80037,
"AF":0, "Zoom":0, "_id_Marca":1},
        {"_id":1, "Modello":"elsqSrUCqpPmvwtNgbRkgHkW",
"OIS":94480, "AF":0, "Zoom":1, "_id_Marca":1},
        {"_id":2, "Modello":"QRuGmtIPMoXTRx", "OIS":42954,
"AF":0, "Zoom":1, "_id_Marca":2},
        {"_id":3, "Modello":"UXaWwSyYeyOeZaeTHcBIyGihFtVZo",
"OIS":14570, "AF":1, "Zoom":1, "_id_Marca":2},
        {"_id":4,
"Modello":"eIMWIYZXpxTHTQutPzfTKlVhJmugrAJJEpg", "OIS":68348, "AF":0, "Zoom":0,
"_id_Marca":4}
    ])

db.Compatta.insert([
        {"_id_Macchina":1, "_id_Obiettivo":1,
"ZoomDigitale":1, "StabilizzazioneDigitale":1},
        {"_id_Macchina":2, "_id_Obiettivo":2,
"ZoomDigitale":0, "StabilizzazioneDigitale":1}
    ])

db.Reflex.insert([
        {"_id_Macchina":0, "MotoreAF":1},
        {"_id_Macchina":3, "MotoreAF":1},
        {"_id_Macchina":4, "MotoreAF":1}
    ])

```

```

])

db.Monta.insert([
    {"_id_Macchina":0, "_id_Obiettivo":1,
"Attacco":"CULnl"},
    {"_id_Macchina":0, "_id_Obiettivo":2,
"Attacco":"Xm"},
    {"_id_Macchina":3, "_id_Obiettivo":2,
"Attacco":"KIjd"},
    {"_id_Macchina":4, "_id_Obiettivo":3,
"Attacco":"FfIT"}
])

db.Fotografia.insert([
    {"_id":0,
"Descrizione":"JTkQKrHXObLopnJqgbYsliWewUPdte", "Data":"2008-4-14",
"_id_Album":3, "_id_Categoria":1, "_id_Luogo":2, "_id_Macchina":0,
"_id_Obiettivo":1, "LunghezzaFocale":1.7, "Angolo":58, "Diaframma":2.5,
"Esposizione":-284, "ISO":1673, "Risoluzione":2175},
    {"_id":1,
"Descrizione":"zIoufGfZoeQUdXjNkycOthyYijNeBn", "Data":"2006-2-13",
"_id_Album":2, "_id_Categoria":1, "_id_Luogo":3, "_id_Macchina":0,
"_id_Obiettivo":2, "LunghezzaFocale":2.8, "Angolo":73, "Diaframma":2.9,
"Esposizione":-754, "ISO":2610, "Risoluzione":3085},
    {"_id":2,
"Descrizione":"SUgnnWTcmwDtPTXjBXVwClhCQKjoyI", "Data":"1989-3-25",
"_id_Album":2, "_id_Categoria":1, "_id_Luogo":4, "_id_Macchina":3,
"_id_Obiettivo":2, "LunghezzaFocale":1.5, "Angolo":7, "Diaframma":2.6,
"Esposizione":-312, "ISO":979, "Risoluzione":1704},
    {"_id":3,
"Descrizione":"dmWulFCVQULHjmKHfNbvcMFbmeQe", "Data":"2007-9-24",
"_id_Album":3, "_id_Categoria":2, "_id_Luogo":1, "_id_Macchina":4,
"_id_Obiettivo":3, "LunghezzaFocale":2.5, "Angolo":108, "Diaframma":2.4,
"Esposizione":-28, "ISO":4740, "Risoluzione":2040},
    {"_id":4,
"Descrizione":"ODyoxMZlNlHWOAoiWBXAOWowdlaWy", "Data":"2008-9-3", "_id_Album":1,
"_id_Categoria":2, "_id_Luogo":1, "_id_Macchina":2, "_id_Obiettivo":2,
"LunghezzaFocale":1.6, "Angolo":77, "Diaframma":1.8, "Esposizione":-255,
"ISO":3925, "Risoluzione":508}
])

db.Voto.insert([
    {"_id":0, "_id_Utente":3, "_id_Foto":1, "Voto":10,
"Descrizione":"gZHaSAyVevcUNGvqmgmsUKoYhdtTkCgDzSt"},
    {"_id":1, "_id_Utente":4, "_id_Foto":1, "Voto":10,
"Descrizione":"zZFMBCFPxyvGrIEiAryAvBtvVZLyIjlTTckfK"},
    {"_id":2, "_id_Utente":2, "_id_Foto":2, "Voto":3,
"Descrizione":"iQjQuBkPNJzulNEWwRoVXLNiPBdlNsrQwOEUgbHTCQDaTBiyZ"},
    {"_id":3, "_id_Utente":1, "_id_Foto":0, "Voto":1,
"Descrizione":"tkeTwTKJkwDaeIHUkJzuPkjJTbgYSJrV"},
    {"_id":4, "_id_Utente":0, "_id_Foto":3, "Voto":1,
"Descrizione":"yKQeDCbsAezSLgtsDMYLUNCEc"}
])

```

8.2 - Query

Traduzione in MongoDB della query numero 1:

Mostrare il numero di foto per ogni categoria.

- **MYSQL:**

```
SELECT CATEGORIA.Nome, COUNT(*) as NumeroFoto
FROM FOTOGRAFIA, CATEGORIA
WHERE FOTOGRAFIA.ID_Categoria = CATEGORIA.ID
GROUP BY CATEGORIA.Nome;
```

- **MONGODB:**

```
db.Fotografia.aggregate([{"$group":{"_id":"$_id_Categoria",
"count":{"$sum":1}}}]])
```

Traduzione in MongoDB della query numero 2:

Mostrare il numero di utenti registrati raggruppandoli per anno di nascita.

- **MYSQL:**

```
SELECT YEAR(UTENTE.Data) as Anno, COUNT(UTENTE.ID) as NumeroUtenti
FROM UTENTE
GROUP BY YEAR(UTENTE.Data);
```

- **MONGODB:**

```
db.Utente.aggregate({"$group":{"_id": { $substr: [ "$Data", 0, 4 ]},
"count":{"$sum":1}}}]])
```

Traduzione in MongoDB della query numero 3:

Mostrare gli utenti nati dopo il 1995.

- **MYSQL:**

```
SELECT UTENTE.ID, UTENTE.Nome, UTENTE.Cognome, UTENTE.Data
FROM UTENTE
WHERE YEAR(UTENTE.Data) >= 1995;
```

- **MONGODB:**

```
db.Utente.find( { Data : { $regex : /^1995/ } }, { Nome:1, Cognome:1, Data:1 }
)
```

Traduzione in MongoDB della query numero 9:

Stampare gli utenti che hanno il numero di cellulare che inizia per 346 .

- **MYSQL:**

```
SELECT UTENTE.ID, UTENTE.Nome, UTENTE.Cognome, UTENTE.Cellulare
FROM UTENTE
WHERE UTENTE.Cellulare like "334%";
```

- **MONGODB:**

```
db.Utente.find( { Cellulare : { $regex : /^346/ } } , { Nome:1, Cognome:1,
Cellulare:1 } )
```


8.3 - Tempi di Risposta

Query1

```
db.Fotografia.explain("executionStats").aggregate([{"$group":{"_id":"$_id_Categoria", "count":{"$sum":1}}}]])
```

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 100000,
  "executionTimeMillis" : 637,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 100000,
  "executionStages" : {
    "stage" : "COLLSCAN",
    "nReturned" : 100000,
    "executionTimeMillisEstimate" : 27,
    "works" : 100002,
    "advanced" : 100000,
    "needTime" : 1,
    "needYield" : 0,
    "saveState" : 790,
    "restoreState" : 790,
    "isEOF" : 1,
    "direction" : "forward",
    "docsExamined" : 100000
  }
}
```

Query2

```
db.Utente.explain("executionStats").aggregate({"$group":{"_id": { $substr: [ "$Data", 0, 4 ]}, "count":{"$sum":1}}}]])
```

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 100000,
  "executionTimeMillis" : 239,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 100000,
  "executionStages" : {
    "stage" : "COLLSCAN",
    "nReturned" : 100000,
    "executionTimeMillisEstimate" : 2,
    "works" : 100002,
    "advanced" : 100000,
    "needTime" : 1,
    "needYield" : 0,
    "saveState" : 789,
    "restoreState" : 789,
    "isEOF" : 1,
    "direction" : "forward",
    "docsExamined" : 100000
  }
}
```

Query3

```
db.Utente.find( { Data : { $regex : /^1995/} }, { Nome:1, Cognome:1, Data:1} )
```

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 2520,
  "executionTimeMillis" : 115,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 100000,
  "executionStages" : {
    "stage" : "PROJECTION_SIMPLE",
    "nReturned" : 2520,
    "executionTimeMillisEstimate" : 1,
    "works" : 100002,
    "advanced" : 2520,
    "needTime" : 97481,
    "needYield" : 0,
    "saveState" : 781,
    "restoreState" : 781,
    "isEOF" : 1,
    "transformBy" : {
      "Nome" : 1,
      "Cognome" : 1,
      "Data" : 1
    }
  },
  "inputStage" : {
    "stage" : "COLLSCAN",
    "filter" : {
      "Data" : {
        "$regex" : "^1995"
      }
    },
    "nReturned" : 2520,
    "executionTimeMillisEstimate" : 1,
    "works" : 100002,
    "advanced" : 2520,
    "needTime" : 97481,
    "needYield" : 0,
    "saveState" : 781,
    "restoreState" : 781,
    "isEOF" : 1,
    "direction" : "forward",
    "docsExamined" : 100000
  }
},
}
```

Query9

```
db.Utente.find({ Cellulare : { $regex : /^346/} } , {Nome:1, Cognome:1, Cellulare:1} )
```

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 116,
  "executionTimeMillis" : 146,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 100000,
  "executionStages" : {
    "stage" : "PROJECTION_SIMPLE",
    "nReturned" : 116,
    "executionTimeMillisEstimate" : 28,
    "works" : 100002,
    "advanced" : 116,
    "needTime" : 99885,
    "needYield" : 0,
    "saveState" : 781,
    "restoreState" : 781,
    "isEOF" : 1,
    "transformBy" : {
      "Nome" : 1,
      "Cognome" : 1,
      "Cellulare" : 1
    }
  },
  "inputStage" : {
    "stage" : "COLLSCAN",
    "filter" : {
      "Cellulare" : {
        "$regex" : "^334"
      }
    },
    "nReturned" : 116,
    "executionTimeMillisEstimate" : 28,
    "works" : 100002,
    "advanced" : 116,
    "needTime" : 99885,
    "needYield" : 0,
    "saveState" : 781,
    "restoreState" : 781,
    "isEOF" : 1,
    "direction" : "forward",
    "docsExamined" : 100000
  }
},
}
```