

Progetto Database: manutenzione e gestione impianti radio

Analisi dei requisiti

Introduzione progetto

Il database presenta una situazione lavorativa reale, la quale consiste nella manutenzione e gestione degli impianti radio presenti nelle postazioni dislocate in tutta Italia. Tale impiego usufruisce di un lavoratore, dotato di apposito mezzo di trasporto, in grado di raggiungere la postazione che necessita di un componente. Tale operazione può consistere nella sostituzione di un determinato apparato (esempio: trasmettitore, modulatore di frequenza, codificatore) oppure di una stabilita antenna (esempio: antenna internet, antenna usata per la trasmissione delle onde radio). Un'altra tipologia di intervento che può occorrere è la manutenzione degli impianti quando non è possibile effettuare tali operazioni tramite telemetria.

Dizionario dei dati

Tabella: lavoratore

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_lavoratore	Carattere	6	Codice identificativo, PK	Lav000
nome_lavoratore	Carattere	50	Nome lavoratore	Thomas
cognome_lavoratore	Carattere	50	Cognome lavoratore	Mandija
stipendio	Numerico		Stipendio mensile	1500

Tabella: cliente

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_cliente	Carattere	6	Codice identificativo, PK	Clc000
nome_cliente	Carattere	50	Nome cliente	Radio Globo

Tabella: trasporto

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_trasporto	Carattere	6	Codice identificativo, PK	Trp000
marca	Carattere	50	Marca mezzo di trasporto	Fiat
modello	Carattere	50	Modello mezzo di trasporto	Doblò
posti	Numerico		Indica il numero di posti disponibili	3
prezzo	Numerico		Indica il prezzo in euro	20000

Tabella: regione

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_regione	Carattere	6	Codice identificativo, PK	Reg000
nome	Carattere	50	Nome regione	Lazio
num_interventi	Intero		Numero totale interventi effettuati	1

Tabella: intervento

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_intervento	Carattere	6	Codice identificativo, PK	Itv000
id_lavoratore	Carattere	6	Codice identificativo, FK	Lav000
id_cliente	Carattere	6	Codice identificativo, FK	Clt000
id_trasporto	Carattere	6	Codice identificativo, FK	Trp000
id_regione	Carattere	6	Codice identificativo, FK	Reg000
spesa	Numerico		Spesa complessiva intervento	1100
data_intervento	Data		Data intervento	2021-01-15

Tabella: componenti

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_componenti	Carattere	6	Codice identificativo, PK	Cmp000
Id_intervento	Carattere	6	Codice identificativo, FK	Itv000
id_trasmettitore	Carattere	6	Codice identificativo, FK	Trx000
id_modulatore	Carattere	6	Codice identificativo, FK	Mrx000
id_ricevitore	Carattere	6	Codice identificativo, FK	Rrx000
id_antenna	Carattere	6	Codice identificativo, FK	Ant000
id_postazione	Carattere	6	Codice identificativo, FK	Pos000
id_manutenzione	Carattere	6	Codice identificativo, FK	Mnt000

Tabella: trasmettitore

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_trasmettitore	Carattere	6	Codice identificativo, PK	Trx000
modello	Carattere	50	Descrive il modello del trasmettitore	Trsmt-TR94
prezzo	Numerico		Indica il prezzo in euro	200

Tabella: modulatore

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_modulatore	Carattere	6	Codice identificativo, PK	Mrx000
modello	Carattere	50	Descrive il modello del modulatore	Mod-M47
prezzo	Numerico		Indica il prezzo in euro	200

Tabella: ricevitore

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_ricevitore	Carattere	6	Codice identificativo, PK	Rrx000
modello	Carattere	50	Descrive il modello del ricevitore	Ric-R74
prezzo	Numerico		Indica il prezzo in euro	200

Tabella: antenna

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_antenna	Carattere	6	Codice identificativo, PK	Ant000
modello	Carattere	50	Descrive il modello dell'antenna	Ant-A50
prezzo	Numerico		Indica il prezzo in euro	200

Tabella: postazione

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_postazione	Carattere	6	Codice identificativo, PK	Pos000
nome_postazione	Carattere	50	Nome postazione	Postazione Radio Globo
id_regione	Carattere	6	Codice identificativo, FK	Reg000
id_cliente	Carattere	6	Codice identificativo, FK	Clit000

Tabella: manutenzione

Nome attributo	Tipo dato	Grandezza dato	Descrizione	Esempio
id_manutenzione	Carattere	6	Codice identificativo, PK	Mnt000
descrizione	Carattere	200	Descrive la manutenzione effettuata	Riavvio stazione

Glossario termini

Entità	Descrizione	Collegamenti
Lavoratore	Persona assunta da un'azienda addetta alla manutenzione degli impianti radio ed è colui che raggiunge la postazione che necessita di un intervento e lo esegue.	Intervento
Cliente	Rappresenta un ente di trasmissione radio, nonché la figura che richiede l'intervento all'azienda di manutenzione	Intervento
Trasporto	Definisce gli ipotetici mezzi di trasporto utilizzati per raggiungere il luogo che necessita dell'intervento	Intervento
Regione	Definisce la regione italiana da raggiungere per eseguire l'intervento	Intervento
Intervento	Rappresenta un insieme che collega sotto un'unica voce "intervento" le seguenti voci: il lavoratore, il cliente, il mezzo di trasporto e la regione	

Trasmettitore	Il componente che può essere sostituito durante un intervento, necessario al corretto funzionamento della postazione	Componenti
Modulatore	Il componente che può essere sostituito durante un intervento, necessario al corretto funzionamento della postazione	Componenti
Ricevitore	Il componente che può essere sostituito durante un intervento, necessario al corretto funzionamento della postazione	Componenti
Antenna	Il componente che può essere sostituito durante un intervento, necessario al corretto funzionamento della postazione	Componenti
Postazione	Rappresenta la postazione in cui è necessario effettuare l'intervento	Componenti
Manutenzione	Rappresenta una manutenzione ordinaria, dove non è richiesto lo scambio di componenti	Componenti
Componenti	Definisce un insieme che collega sotto un'unica voce intervento le seguenti voci: il trasmettitore, il modulatore, il ricevitore, l'antenna, la postazione e la manutenzione	Intervento

Schema ER logico

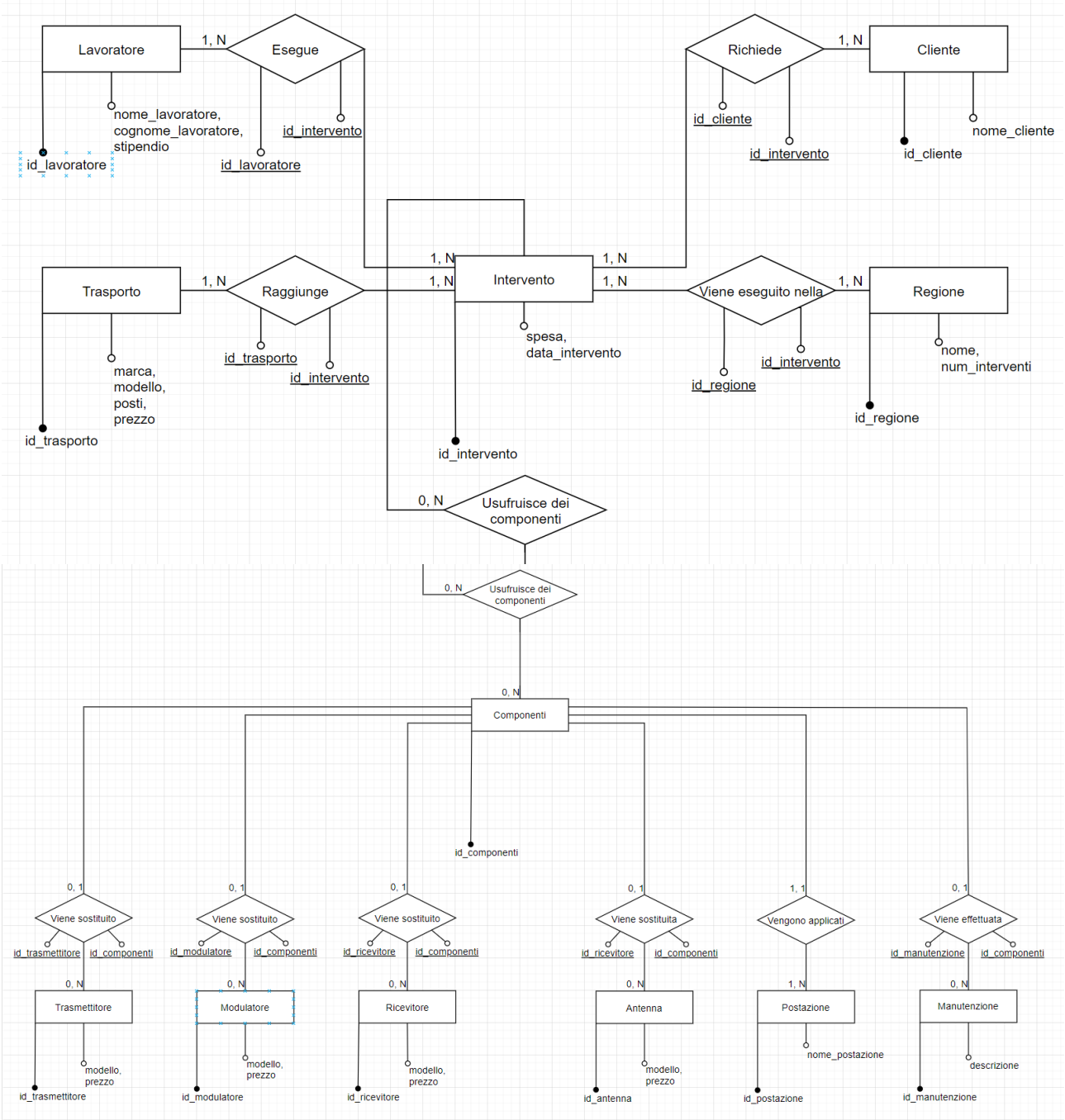
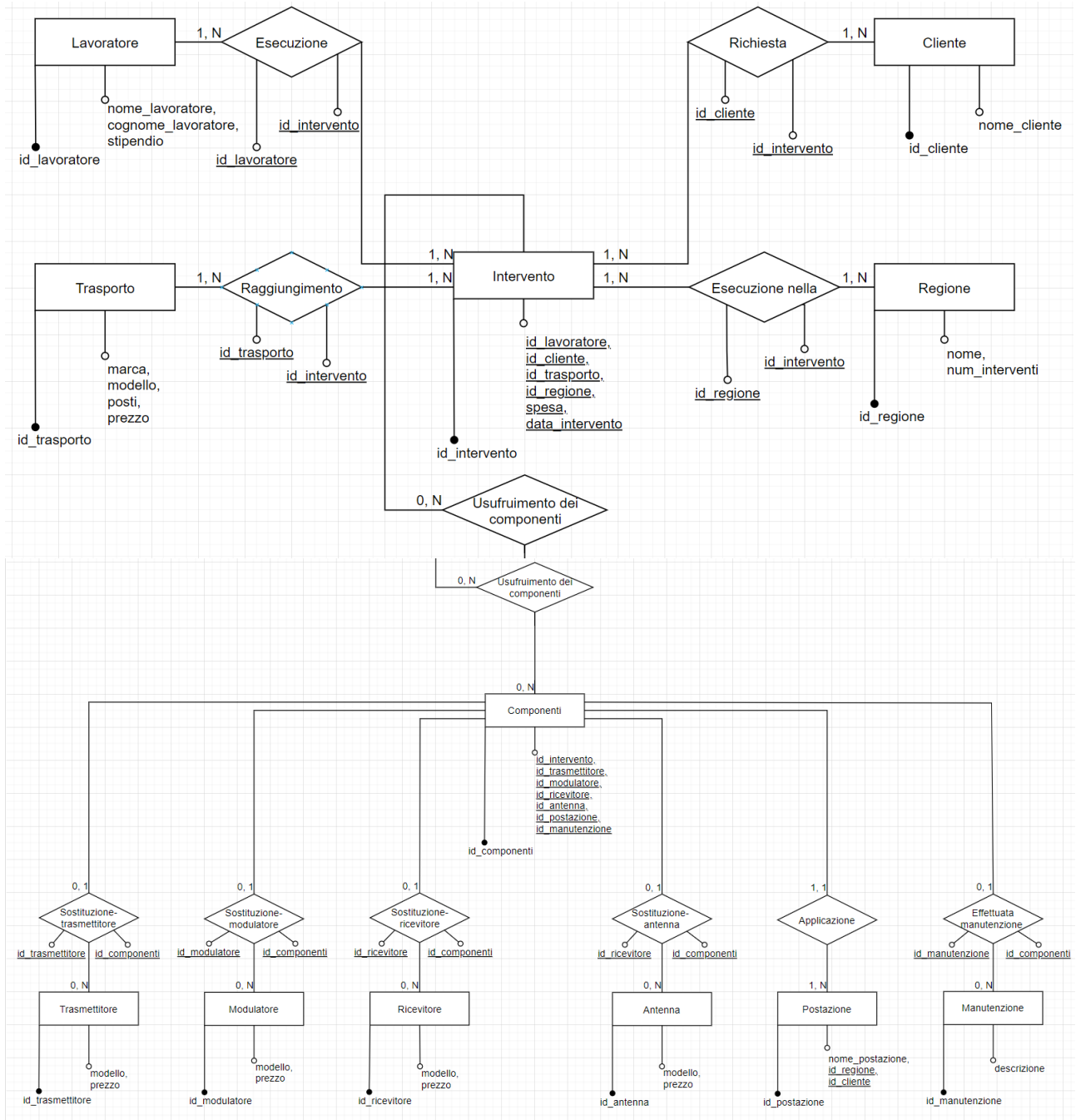


Tabelle Entità

Nome entità	Chiave primaria (PK)	In relazione con	Tramite Chiave Esterna (FK)
Lavoratore	id_lavoratore	Intervento	lavoratore.id_lavoratore = intervento.id_lavoratore
Cliente	id_cliente	Intervento	cliente.id_cliente = intervento.id_cliente
Trasporto	id_trasporto	Intervento	trasporto.id_trasporto = intervento.id_trasporto
Regione	id_regione	Intervento	regione.id_regione = intervento.id_regione
Intervento	id_intervento		
Trasmettitore	id_trasmettitore	Componenti	trasmettitore.id_trasmettitore = componenti.id_trasmettitore
Modulatore	id_modulatore	Componenti	modulatore.id_modulatore = componenti.id_modulatore
Ricevitore	id_ricevitore	Componenti	ricevitore.id_ricevitore = componenti.id_ricevitore
Antenna	id_antenna	Componenti	antenna.id_antenna = componenti.id_antenna
Postazione	id_postazione	Componenti	postazione.id_postazione = componenti.id_postazione
Manutenzione	id_manutenzione	Componenti	manutenzione.id_manutenzione = componenti.id_manutenzione
Componenti	id_intervento, id_trasmettitore, id_modulatore, id_ricevitore, id_antenna, id_postazione, id_manutenzione	Intervento	intervento.id_intervento = componenti.id_intervento

Schema ER fisico



Codice SQL

Creazione tabelle

```
create database radio;
use radio;

create table lavoratore(
    id_lavoratore varchar(6) not null primary key,
    nome_lavoratore varchar(50) not null,
    cognome_lavoratore varchar(50) not null,
    stipendio int not null
);

create table cliente(
    id_cliente varchar(6) not null primary key,
    nome_cliente varchar(50) not null
);

create table trasporto(
    id_trasporto varchar(6) not null primary key,
    marca varchar(50) not null,
    modello varchar(50) not null,
    posti int not null,
    prezzo int not null
);

create table regione(
    id_regione varchar(6) not null primary key,
    nome varchar(50) not null,
    num_interventi int not null
);

create table trasmettitore(
    id_trasmettitore varchar(6) not null primary key,
    modello varchar(50) not null,
    prezzo int not null
);

create table modulatore(
    id_modulatore varchar(6) not null primary key,
    modello varchar(50) not null,
    prezzo int not null
);

create table ricevitore(
    id_ricevitore varchar(6) not null primary key,
    modello varchar(50) not null,
    prezzo int not null
);
```

```

create table antenna(
    id_antenna varchar(6) not null primary key,
    modello varchar(50) not null,
    prezzo int not null
);

create table postazione(
    id_postazione varchar(6) not null primary key,
    nome_postazione varchar(50) not null,
    id_regione varchar(6) not null references regione(id_regione) on update cascade
on delete no action,
    id_cliente varchar(6) not null references cliente(id_cliente) on update cascade
on delete no action
);

create table manutenzione(
    id_manutenzione varchar(6) not null primary key,
    descrizione varchar(200)
);

create table intervento(
    id_intervento varchar(6) not null primary key,
    id_lavoratore varchar(6) not null references lavoratore(id_lavoratore) on update
e cascade on delete no action,
    id_cliente varchar(6) not null references cliente(id_cliente) on update cascade
on delete no action,
    id_trasporto varchar(6) not null references trasporto(id_trasporto) on update c
ascade on delete no action,
    id_regione varchar(6) not null references regione(id_regione) on update cascade
on delete no action,
    spesa int not null,
    data_intervento date not null
);

create table componenti(
    id_componenti varchar(6) not null primary key,
    id_intervento varchar(6) not null references intervento(id_intervento) on updat
e cascade on delete no action,
    id_trasmettitore varchar(6) references trasmettitore(id_trasmettitore) on updat
e cascade on delete no action,
    id_modulatore varchar(6) references modulatore(id_modulatore) on update cascade
on delete no action,
    id_ricevitore varchar(6) references ricevitore(id_ricevitore) on update cascade
on delete no action,
    id_antenna varchar(6) references antenna(id_antenna) on update cascade on delet
e no action,
    id_postazione varchar(6)not null references postazione(id_postazione) on update
cascade on delete no action,
    id_manutenzione varchar(6) references manutenzione(id_manutenzione) on update c
ascade on delete no action
);

```

```
create table esecuzione(  
    id_lavoratore varchar(6) not null references lavoratore(id_lavoratore) on update cascade on delete no action,  
    id_intervento varchar(6) not null references intervento(id_intervento) on update cascade on delete no action  
);  
  
alter table esecuzione  
add primary key (id_lavoratore, id_intervento);  
  
create table richiesta(  
    id_cliente varchar(6) not null references cliente(id_cliente) on update cascade on delete no action,  
    id_intervento varchar(6) not null references intervento(id_intervento) on update cascade on delete no action  
);  
  
alter table richiesta  
add primary key (id_cliente, id_intervento);  
  
create table raggiungimento(  
    id_trasporto varchar(6) not null references trasporto(id_trasporto) on update cascade on delete no action,  
    id_intervento varchar(6) not null references intervento(id_intervento) on update cascade on delete no action  
);  
  
alter table raggiungimento  
add primary key (id_trasporto, id_intervento);  
  
create table esecuzione_reg(  
    id_regione varchar(6) not null references regione(id_regione) on update cascade on delete no action,  
    id_intervento varchar(6) not null references intervento(id_intervento) on update cascade on delete no action  
);  
  
alter table esecuzione_reg  
add primary key (id_regione, id_intervento);  
  
create table sost_trx(  
    id_trasmettitore varchar(6) not null references trasmettitore(id_trasmettitore) on update cascade on delete no action,  
    id_componenti varchar(6) not null references componenti(id_componenti) on update cascade on delete no action  
);  
  
alter table sost_trx  
add primary key (id_trasmettitore, id_componenti);
```

```
create table sost_mrx(  
    id_modulatore varchar(6) not null references modulatore(id_modulatore) on update cascade on delete no action,  
    id_componenti varchar(6) not null references componenti(id_componenti) on update cascade on delete no action  
);  
  
alter table sost_mrx  
add primary key (id_modulatore, id_componenti);  
  
create table sost_rrx(  
    id_ricevitore varchar(6) not null references ricevitore(id_ricevitore) on update cascade on delete no action,  
    id_componenti varchar(6) not null references componenti(id_componenti) on update cascade on delete no action  
);  
  
alter table sost_rrx  
add primary key (id_ricevitore, id_componenti);  
  
create table sost_ant(  
    id_antenna varchar(6) not null references antenna(id_antenna) on update cascade on delete no action,  
    id_componenti varchar(6) not null references componenti(id_componenti) on update cascade on delete no action  
);  
  
alter table sost_ant  
add primary key (id_antenna, id_componenti);  
  
create table sost_mnt(  
    id_manutenzione varchar(6) not null references manutenzione(id_manutenzione) on update cascade on delete no action,  
    id_componenti varchar(6) not null references componenti(id_componenti) on update cascade on delete no action  
);  
  
alter table sost_mnt  
add primary key (id_manutenzione, id_componenti);
```

Inserimento valori tabelle

```
insert into lavoratore
values
('Lav000', 'Thomas', 'Mandija', 1500),
('Lav001', 'Dario', 'Lemeni', 1200),
('Lav002', 'Marco', 'Fei', 1300),
('Lav003', 'Daniele', 'Mei', 800),
('Lav004', 'Enrico', 'Giordani', 1100),
('Lav005', 'Adriano', 'Semerano', 1400),
('Lav006', 'Lorenzo', 'Rocchi', 1000),
('Lav007', 'Francesco', 'Stefani', 1000),
('Lav008', 'Adrian', 'Alexa', 1000),
('Lav009', 'Guido', 'Parlatore', 1000);

insert into cliente
values
('Cl000', 'Radio Globo'),
('Cl001', 'Virgin Radio'),
('Cl002', 'Radio 105'),
('Cl003', 'R101'),
('Cl004', 'Rai1'),
('Cl005', 'Radio Montecarlo'),
('Cl006', 'Radio Radicale'),
('Cl007', 'RTL'),
('Cl008', 'Radio M100'),
('Cl009', 'RDS');

insert into trasporto
values
('Trp000', 'Volkswagen', 'Crafer', 3, 27000),
('Trp001', 'Fiat', 'Doblò', 3, 20000),
('Trp002', 'Peugeot', 'Boxer', 3, 18000),
('Trp003', 'Ford', 'Transit', 3, 27000),
('Trp004', 'Dacia', 'Dokker', 2, 14000),
('Trp005', 'Citroen', 'Berlingo', 3, 15000),
('Trp006', 'Opel', 'Combo', 2, 14500),
('Trp007', 'Mercedes', 'Citan', 3, 18000),
('Trp008', 'Iveco', 'Daily furgone', 3, 35000),
('Trp009', 'Toyota', 'Proace', 3, 23000);
```

```
insert into regione
```

```
values
```

```
('Reg000', 'Lazio', 1),  
( 'Reg001', 'Toscana', 1),  
( 'Reg002', 'Campania', 2),  
( 'Reg003', 'Umbria', 1),  
( 'Reg004', 'Basilicata', 1),  
( 'Reg005', 'Puglia', 1),  
( 'Reg006', 'Sicilia', 1),  
( 'Reg007', 'Sardegna', 0),  
( 'Reg008', 'Lombardia', 1),  
( 'Reg009', 'Piemonte', 1);
```

```
insert into trasmettitore
```

```
values
```

```
('Trx000', 'Trsmt-TR94', 200),  
( 'Trx001', 'Trsmt-TR88', 100),  
( 'Trx002', 'Trsmt-TR12', 300),  
( 'Trx003', 'Trsmt-TR37', 900),  
( 'Trx004', 'Trsmt-TR28', 400),  
( 'Trx005', 'Trsmt-TR71', 500),  
( 'Trx006', 'Trsmt-TR56', 800),  
( 'Trx007', 'Trsmt-TR46', 700),  
( 'Trx008', 'Trsmt-TR77', 600),  
( 'Trx009', 'Trsmt-TR61', 1000);
```

```
insert into modulatore
```

```
values
```

```
('Mrx000', 'Mod-M47', 200),  
( 'Mrx001', 'Mod-M56', 200),  
( 'Mrx002', 'Mod-M18', 200),  
( 'Mrx003', 'Mod-M43', 150),  
( 'Mrx004', 'Mod-M98', 150),  
( 'Mrx005', 'Mod-M12', 150),  
( 'Mrx006', 'Mod-M78', 500),  
( 'Mrx007', 'Mod-M52', 300),  
( 'Mrx008', 'Mod-M91', 100),  
( 'Mrx009', 'Mod-M64', 700);
```

```
insert into ricevitore
```

```
values
```

```
('Rrx000', 'Ric-R74', 200),  
( 'Rrx001', 'Ric-R53', 200),  
( 'Rrx002', 'Ric-R69', 200),  
( 'Rrx003', 'Ric-R78', 450),  
( 'Rrx004', 'Ric-R64', 900),  
( 'Rrx005', 'Ric-R42', 300),  
( 'Rrx006', 'Ric-R11', 300),  
( 'Rrx007', 'Ric-R94', 600),
```

```
('Rrx008', 'Ric-R63', 600),  
( 'Rrx009', 'Ric-R83', 600);
```

```
insert into antenna
```

```
values
```

```
('Ant000', 'Ant-A60', 100),  
( 'Ant001', 'Ant-A50', 500),  
( 'Ant002', 'Ant-A80', 700),  
( 'Ant003', 'Ant-A40', 200),  
( 'Ant004', 'Ant-A70', 400),  
( 'Ant005', 'Ant-A20', 600),  
( 'Ant006', 'Ant-A10', 800),  
( 'Ant007', 'Ant-A30', 1000),  
( 'Ant008', 'Ant-A90', 300),  
( 'Ant009', 'Ant-A00', 900);
```

```
insert into postazione
```

```
values
```

```
('Pos000', 'Pos0', 'Reg000', 'Clt000'),  
( 'Pos001', 'Pos1', 'Reg002', 'Clt002'),  
( 'Pos002', 'Pos2', 'Reg001', 'Clt001'),  
( 'Pos003', 'Pos3', 'Reg005', 'Clt005'),  
( 'Pos004', 'Pos4', 'Reg004', 'Clt009'),  
( 'Pos005', 'Pos5', 'Reg006', 'Clt004'),  
( 'Pos006', 'Pos6', 'Reg008', 'Clt006'),  
( 'Pos007', 'Pos7', 'Reg007', 'Clt008'),  
( 'Pos008', 'Pos8', 'Reg009', 'Clt007'),  
( 'Pos009', 'Pos9', 'Reg003', 'Clt003');
```

```
insert into manutenzione
```

```
values
```

```
('Mnt000', 'Riavvio stazione'),  
( 'Mnt001', 'Spegnimento completo, pulizia sistemi'),  
( 'Mnt002', 'Cambio gruppo elettrogeno'),  
( 'Mnt003', 'Controllo stato'),  
( 'Mnt004', 'Sostituzione batterie');
```

```
insert into intervento
```

```
values
```

```
('Itv000', 'Lav000', 'Clt000', 'Trp002', 'Reg000', 0, '2021-01-15'),  
( 'Itv001', 'Lav002', 'Clt001', 'Trp001', 'Reg001', 0, '2021-01-16'),  
( 'Itv002', 'Lav001', 'Clt002', 'Trp000', 'Reg002', 0, '2020-04-25'),  
( 'Itv003', 'Lav001', 'Clt002', 'Trp001', 'Reg002', 0, '2021-08-05'),  
( 'Itv004', 'Lav005', 'Clt006', 'Trp000', 'Reg008', 0, '2021-06-03'),  
( 'Itv005', 'Lav008', 'Clt003', 'Trp003', 'Reg003', 0, '2020-01-08'),  
( 'Itv006', 'Lav007', 'Clt005', 'Trp005', 'Reg005', 0, '2020-02-12'),  
( 'Itv007', 'Lav006', 'Clt009', 'Trp007', 'Reg004', 0, '2021-05-23'),  
( 'Itv008', 'Lav004', 'Clt004', 'Trp006', 'Reg006', 0, '2021-01-25'),  
( 'Itv009', 'Lav009', 'Clt007', 'Trp008', 'Reg009', 0, '2021-06-01');
```

```
insert into componenti
values
('Cmp000', 'Itv000', 'Trx000', 'Mrx001', 'Rrx002', 'Ant001', 'Pos000', NULL),
('Cmp001', 'Itv001', NULL, 'Mrx002', NULL, 'Ant002', 'Pos002', NULL),
('Cmp002', 'Itv002', NULL, NULL, NULL, NULL, 'Pos001', 'Mnt000'),
('Cmp003', 'Itv003', NULL, 'Mrx002', NULL, NULL, 'Pos001', NULL),
('Cmp004', 'Itv004', 'Trx008', 'Mrx001', 'Rrx008', 'Ant005', 'Pos006', NULL),
('Cmp005', 'Itv005', NULL, NULL, NULL, NULL, 'Pos009', 'Mnt001'),
('Cmp006', 'Itv006', NULL, NULL, NULL, NULL, 'Pos003', 'Mnt004'),
('Cmp007', 'Itv007', 'Trx003', 'Mrx006', NULL, 'Ant008', 'Pos004', NULL),
('Cmp008', 'Itv008', 'Trx007', 'Mrx006', NULL, NULL, 'Pos005', NULL),
('Cmp009', 'Itv009', 'Trx006', 'Mrx006', 'Rrx003', 'Ant006', 'Pos008', NULL);
```

```
insert into sost_trx
values
```

```
('Trx000', 'Cmp000'),
('Trx008', 'Cmp004'),
('Trx003', 'Cmp007'),
('Trx007', 'Cmp006'),
('Trx006', 'Cmp009');
```

```
insert into sost_mrx
values
```

```
('Mrx001', 'Cmp000'),
('Mrx002', 'Cmp001'),
('Mrx002', 'Cmp003'),
('Mrx001', 'Cmp004'),
('Mrx006', 'Cmp007'),
('Mrx006', 'Cmp008'),
('Mrx006', 'Cmp009');
```

```
insert into sost_rrx
values
```

```
('Rrx002', 'Cmp000'),
('Rrx008', 'Cmp004'),
('Rrx003', 'Cmp009');
```

```
insert into sost_ant
values
```

```
('Ant001', 'Cmp000'),
('Ant002', 'Cmp001'),
('Ant005', 'Cmp004'),
('Ant008', 'Cmp007'),
('Ant006', 'Cmp009');
```

```
insert into sost_mnt
values
```

```
('Mnt000', 'Cmp002'),
('Mnt001', 'Cmp005'),
('Mnt004', 'Cmp006');
```



```
insert into esecuzione
values
```

```
('Lav000', 'Itv000'),
('Lav001', 'Itv002'),
('Lav001', 'Itv003'),
('Lav002', 'Itv001'),
('Lav004', 'Itv008'),
('Lav005', 'Itv004'),
('Lav006', 'Itv007'),
('Lav007', 'Itv006'),
('Lav008', 'Itv005'),
('Lav009', 'Itv009');
```

```
insert into richiesta
values
```

```
('Clt000', 'Itv000'),
('Clt001', 'Itv001'),
('Clt002', 'Itv002'),
('Clt002', 'Itv003'),
('Clt006', 'Itv004'),
('Clt003', 'Itv005'),
('Clt005', 'Itv006'),
('Clt009', 'Itv007'),
('Clt004', 'Itv008'),
('Clt007', 'Itv009');
```

```
insert into raggiungimento
values
```

```
('Trp002', 'Itv000'),
('Trp001', 'Itv001'),
('Trp000', 'Itv002'),
('Trp001', 'Itv003'),
('Trp000', 'Itv004'),
('Trp003', 'Itv005'),
('Trp005', 'Itv006'),
('Trp007', 'Itv007'),
('Trp006', 'Itv008'),
('Trp008', 'Itv009');
```

```
insert into esecuzione_reg
values
('Reg000', 'Itv000'),
('Reg001', 'Itv001'),
('Reg002', 'Itv002'),
('Reg002', 'Itv003'),
('Reg008', 'Itv004'),
('Reg003', 'Itv005'),
('Reg005', 'Itv006'),
('Reg004', 'Itv007'),
('Reg006', 'Itv008'),
('Reg009', 'Itv009');
```

Query SQL

```
1. visualizzare il lavoratore, il mezzo di trasporto e la postazione raggiunta
select lavoratore.nome_lavoratore, lavoratore.cognome_lavoratore, trasporto.marca,
trasporto.modello, postazione.nome_postazione
from interv
inner join lavoratore on lavoratore.id_lavoratore = interv.id_lavoratore
inner join trasporto on trasporto.id_trasporto = interv.id_trasporto
inner join postazione on postazione.id_postazione = interv.id_postazione
order by interv.id_postazione;

2. visualizzare il cliente, la regione e la postazione adibita
select cliente.nome_cliente, regione.nome, postazione.nome_postazione
from interv
inner join cliente on cliente.id_cliente = interv.id_cliente
inner join regione on regione.id_regione = interv.id_regione
inner join postazione on postazione.id_postazione = interv.id_postazione
order by cliente.id_cliente;

3. visualizzare il lavoratore, il cliente e i componenti utilizzati durante l'intervento
select lavoratore.nome_lavoratore, lavoratore.cognome_lavoratore, cliente.nome_cliente,
interv.id_intervento, trasmettitore.modello as trasmettitore, modulatore.modello as modulatore,
ricevitore.modello as ricevitore, antenna.modello as antenna
from interv
inner join lavoratore on interv.id_lavoratore = lavoratore.id_lavoratore
inner join cliente on interv.id_cliente = cliente.id_cliente
left join trasmettitore on interv.id_trasmettitore = trasmettitore.id_trasmettitore
left join modulatore on interv.id_modulatore = modulatore.id_modulatore
left join ricevitore on interv.id_ricevitore = ricevitore.id_ricevitore
left join antenna on interv.id_antenna = antenna.id_antenna;
```

4. visualizzare il lavoratore, la postazione e la manutenzione effettuata

```
select lavoratore.nome_lavoratore, lavoratore.cognome_lavoratore, postazione.nome_postazione, manutenzione.descrizione
from interv
inner join lavoratore on interv.id_lavoratore = lavoratore.id_lavoratore
inner join postazione on interv.id_postazione = postazione.id_postazione
inner join manutenzione on interv.id_manutenzione = manutenzione.id_manutenzione;
```

5. visualizzare il cliente, il modello e la marca del mezzo di trasporto ed infine il modello dei componenti

```
select interv.id_intervento, cliente.nome_cliente, trasporto.marca, trasporto.modello, trasmettitore.modello as trasmettitore, modulatore.modello as modulatore, ricevitore.modello as ricevitore, antenna.modello as antenna
from interv
left join trasmettitore on interv.id_trasmettitore = trasmettitore.id_trasmettitore
left join cliente on interv.id_cliente = cliente.id_cliente
left join trasporto on interv.id_trasporto = trasporto.id_trasporto
left join modulatore on interv.id_modulatore = modulatore.id_modulatore
left join ricevitore on interv.id_ricevitore = ricevitore.id_ricevitore
left join antenna on interv.id_antenna = antenna.id_antenna;
```

6. visualizzare l'intervento con spesa minima e lo stipendio del lavoratore

```
select intervento.id_intervento, intervento.spesa, lavoratore.id_lavoratore, lavoratore.stipendio
from intervento
inner join lavoratore on intervento.id_lavoratore = lavoratore.id_lavoratore
where intervento.spesa = 0;
```

7. visualizzare la regione, il cliente e la data dell'intervento

```
select regione.nome, cliente.nome_cliente, intervento.data_intervento
from intervento
inner join regione on regione.id_regione = intervento.id_regione
inner join cliente on cliente.id_cliente = intervento.id_cliente
order by intervento.data_intervento;
```

8. visualizzare gli interventi effettuati nel mese di gennaio

```
select intervento.id_intervento, intervento.data_intervento
from intervento
where data_intervento >= '2021-01-01' and data_intervento <= '2021-01-31';
```

9. visualizzare il cliente, la regione, la postazione e il numero di interventi effettuati

```
select cliente.nome_cliente, regione.nome, postazione.id_postazione, count(postazione.id_postazione) as 'numero interventi'
from interv
inner join postazione on postazione.id_postazione = interv.id_postazione
inner join cliente on cliente.id_cliente = interv.id_cliente
inner join regione on regione.id_regione = interv.id_regione
group by postazione.id_postazione
order by postazione.id_postazione;
```

10. visualizzare i costi di tutti gli interventi effettuati nel 2021

```
select intervento.id_intervento, intervento.spesa, intervento.data_intervento
from intervento
where intervento.data_intervento >= '2021-01-01'
and intervento.data_intervento <= '2021-12-31';
```

11. calcolare il costo totale dei componenti utilizzati durante il primo intervento

```
select interv.id_intervento, interv.id_componenti, sum(coalesce(trasmettitore.prezzo, 0) + coalesce(modulatore.prezzo, 0) + coalesce(ricevitore.prezzo, 0) + coalesce(antenna.prezzo, 0)) as 'costo intervento'
from interv
left join trasmettitore on trasmettitore.id_trasmettitore = interv.id_trasmettitore
left join modulatore on modulatore.id_modulatore = interv.id_modulatore
left join ricevitore on ricevitore.id_ricevitore = interv.id_ricevitore
left join antenna on antenna.id_antenna = interv.id_antenna
where interv.id_intervento = 'Itv000';
```

12. calcolare il costo totale dei componenti utilizzati in tutte le postazioni della regione Lazio

```
select interv.id_intervento, regione.nome, interv.id_postazione, trasmettitore.prezzo as 'prezzo trx', modulatore.prezzo as 'prezzo mrx', ricevitore.prezzo as 'prezzo rrx', antenna.prezzo as 'prezzo ant'
from interv
left join trasmettitore on trasmettitore.id_trasmettitore = interv.id_trasmettitore
left join modulatore on modulatore.id_modulatore = interv.id_modulatore
left join ricevitore on ricevitore.id_ricevitore = interv.id_ricevitore
left join antenna on antenna.id_antenna = interv.id_antenna
inner join regione on regione.id_regione = interv.id_regione
where regione.nome = 'Lazio';
```

13. calcolare il costo totale dei modulatori richiesti per conto del cliente Radio 105

```
select interv.id_intervento, cliente.nome_cliente, modulatore.modello, modulatore.prezzo
from interv
inner join modulatore on modulatore.id_modulatore = interv.id_modulatore
inner join cliente on cliente.id_cliente = interv.id_cliente
where cliente.nome_cliente = 'Radio 105';
```

14. visualizzare quante volte è stato utilizzato lo stesso modulatore durante tutti gli interventi

```
select interv.id_lavoratore, interv.id_intervento, interv.id_modulatore, modulatore.modello
from interv
inner join modulatore on modulatore.id_modulatore = interv.id_modulatore
order by interv.id_modulatore;
```

15. visualizzare i mezzi di trasporto utilizzati per raggiungere le postazioni nella regione Toscana

```
select trasporto.modelo, regione.nome, postazione.id_postazione
from trasporto
inner join intervento on intervento.id_trasporto = trasporto.id_trasporto
inner join regione on regione.id_regione = intervento.id_regione
inner join postazione on postazione.id_regione = regione.id_regione
where regione.nome = 'Toscana';
```

16. visualizzare il lavoratore e gli interventi effettuati per conto del cliente Virgin Radio

```
select lavoratore.id_lavoratore, intervento.id_intervento, cliente.nome_cliente
from lavoratore
inner join intervento on intervento.id_lavoratore = lavoratore.id_lavoratore
inner join cliente on intervento.id_cliente = cliente.id_cliente
where cliente.nome_cliente = 'Virgin Radio';
```

17. visualizzare i componenti utilizzati nelle postazioni del cliente Radio Globo

```
select postazione.nome_postazione, cliente.nome_cliente, componenti.id_intervento,
trasmettitore.modelo, modulatore.modelo, ricevitore.modelo, antenna.modelo
from componenti
right join trasmettitore on componenti.id_trasmettitore = trasmettitore.id_trasmettitore
right join modulatore on componenti.id_modulatore = modulatore.id_modulatore
right join ricevitore on componenti.id_ricevitore = ricevitore.id_ricevitore
right join antenna on componenti.id_antenna = antenna.id_antenna
inner join postazione on componenti.id_postazione = postazione.id_postazione
inner join cliente on postazione.id_cliente = cliente.id_cliente
where cliente.nome_cliente = 'Radio Globo';
```

18. visualizzare tutti gli interventi in cui è stato utilizzato il trasmettitore Trsmt-TR94

```
select interv.id_intervento, trasmettitore.id_trasmettitore, trasmettitore.modelo
from interv
inner join trasmettitore on trasmettitore.id_trasmettitore = interv.id_trasmettitore
where trasmettitore.modelo = 'Trsmt-TR94';
```

19. visualizzare tutti gli interventi in cui è stata effettuata la manutenzione nella regione Toscana

```
select interv.id_intervento, regione.nome, manutenzione.id_manutenzione, manutenzione.descrizione
from interv
inner join manutenzione on manutenzione.id_manutenzione = interv.id_manutenzione
inner join regione on regione.id_regione = interv.id_regione
where regione.nome = 'Campania';
```

20. visualizzare il lavoratore e gli interventi in cui è stata sostituita l'antenna Ant-A50

```
select lavoratore.nome_lavoratore, lavoratore.cognome_lavoratore, interv.id_intervento, antenna.modello
from interv
inner join lavoratore on lavoratore.id_lavoratore = interv.id_lavoratore
inner join antenna on antenna.id_antenna = interv.id_antenna
where antenna.modello = 'Ant-A50';
```

Query algebra relazionale

1. visualizzare l'intervento con spesa minima e lo stipendio del lavoratore

```
select intervento.id_intervento, intervento.spesa, lavoratore.stipendio
from intervento
inner join lavoratore on intervento.id_lavoratore = lavoratore.id_lavoratore
where intervento.spesa = 0;
```

π intervento.id_intervento, intervento.spesa, lavoratore.stipendio (intervento \bowtie (intervento.id_lavoratore = lavoratore.id_lavoratore) lavoratore σ (intervento.spesa = 0))

2. visualizzare gli interventi effettuati nel mese di gennaio

```
select intervento.id_intervento, intervento.data_intervento
from intervento
where data_intervento >= '2021-01-01' and data_intervento <= '2021-01-31';
```

π id_intervento, data_intervento (σ data_intervento >= '2021-01-01' and data_intervento <= '2021-01-31' (intervento))

Viste

collegamento tabella intervento e componenti

```
create view interv as
select intervento.* , componenti.id_componenti, componenti.id_trasmettitore, componenti.id_modulatore, componenti.id_ricevitore, componenti.id_antenna, componenti.id_postazione, componenti.id_manutenzione
from intervento
inner join componenti on intervento.id_intervento = componenti.id_intervento;
```

Trigger

```
trigger stipendio minimo
delimiter //
    create trigger stipendio
    before insert
    on lavoratore
    for each row
    begin
    if (new.stipendio < 1000) then
    set new.stipendio = 1000;
    end if;
    end; //
delimiter ;
```

Store Procedure

```
store procedure per calcolare la spesa dei componenti e salva il valore
create procedure spesa_interv (IN codice varchar(6), OUT spesa int)
select sum(coalesce(trasmettitore.prezzo, 0) + coalesce(modulatore.prezzo, 0) + coa
lesce(ricevitore.prezzo, 0) + coalesce(antenna.prezzo, 0) ) as 'costo intervento'
into spesa
from interv
left join trasmettitore on trasmettitore.id_trasmettitore = interv.id_trasmettitore
left join modulatore on modulatore.id_modulatore = interv.id_modulatore
left join ricevitore on ricevitore.id_ricevitore = interv.id_ricevitore
left join antenna on antenna.id_antenna = interv.id_antenna
where interv.id_intervento = codice;

create procedure up_spesa (IN codice varchar(6))
update intervento
set intervento.spesa = @spesa
where intervento.id_intervento = codice;

call spesa_interv('Itv001', @spesa);
select @spesa;

call up_spesa('Itv001');
```

MongoDB

Scelta tabelle SQL

```
create database radio2;
use radio2;

create table lavoratoreMongoDB(
    id_lavoratore varchar(9) not null primary key,
    nome_lavoratore varchar(50) not null,
    cognome_lavoratore varchar(50) not null
);

create table clienteMongoDB(
    id_cliente varchar(9) not null primary key,
    nome_cliente varchar(50) not null
);

create table trasportoMongoDB(
    id_trasporto varchar(9) not null primary key,
    marca varchar(50) not null,
    modello varchar(50) not null
);

create table regioneMongoDB(
    id_regione varchar(9) not null primary key,
    nome_regione varchar(50) not null
);

create table interventoMongoDB(
    id_intervento varchar(9) not null primary key,
    id_lavoratore varchar(9) not null references lavoratore(id_lavoratore) on update cascade on delete no action,
    id_cliente varchar(9) not null references cliente(id_cliente) on update cascade on delete no action,
    id_trasporto varchar(9) not null references trasporto(id_trasporto) on update cascade on delete no action,
    id_regione varchar(9) not null references regione(id_regione) on update cascade on delete no action
);
```


Generazione valori casuali script Python

```
file = open('MongoDB/File SQL 200k/Mandija_Intervento_200k.sql', 'w') # crea un file TXT in scrittura

file.write('insert into interventoMongoDB values \n') # scrivo nel file TXT la stringa

for x in range (1, 10): # genera i valori che vanno da 1 a 9
    file.write("('Itv00000'+str(x)+'', 'Lav00000'+str(x)+'', 'Clit00000'+str(x)+'', 'Trp00000'+str(x)+'', 'Reg00000'+str(x)+'')," + "\n")

for x in range (10, 100): # genera i valori che vanno da 10 a 99
    file.write("('Itv0000'+str(x)+'', 'Lav0000'+str(x)+'', 'Clit0000'+str(x)+'', 'Trp0000'+str(x)+'', 'Reg0000'+str(x)+'')," + "\n")

for x in range (100, 1000): # genera i valori che vanno da 100 a 999
    file.write("('Itv000'+str(x)+'', 'Lav000'+str(x)+'', 'Clit000'+str(x)+'', 'Trp000'+str(x)+'', 'Reg000'+str(x)+'')," + "\n")

for x in range (1000, 10000): # genera i valori che vanno da 1000 a 9999
    file.write("('Itv00'+str(x)+'', 'Lav00'+str(x)+'', 'Clit00'+str(x)+'', 'Trp00'+str(x)+'', 'Reg00'+str(x)+'')," + "\n")

for x in range (10000, 100000): # genera i valori che vanno da 10000 a 99999
    file.write("('Itv0'+str(x)+'', 'Lav0'+str(x)+'', 'Clit0'+str(x)+'', 'Trp0'+str(x)+'', 'Reg0'+str(x)+'')," + "\n")

for x in range (100000, 200001): # genera i valori che vanno da 100000 a 200000
    if x == 200000:
        file.write("('Itv'+str(x)+'', 'Lav'+str(x)+'', 'Clit'+str(x)+'', 'Trp'+str(x)+'', 'Reg'+str(x)+'');" + "\n")
    else:
        file.write("('Itv'+str(x)+'', 'Lav'+str(x)+'', 'Clit'+str(x)+'', 'Trp'+str(x)+'', 'Reg'+str(x)+'')," + "\n")

print("Scrittura Intervento terminata") # messaggio a schermo
file.close() # chiude il file

#-----
-----#
```

```

file = open('MongoDB/File SQL 200k/Mandija_Lavoratore_200k.sql', 'w') # crea un f
ile TXT in scrittura
file.write('insert into lavoratoreMongoDB values \n') # scrivo ne
l file TXT la stringa

for x in range (1, 10): # genera i
valori che vanno da 1 a 9
    file.write("('Lav00000'+str(x)+'', 'Nome-Lavoratore00000'+str(x)+'', 'Cognome-
Lavoratore00000'+str(x)+''), \n")

for x in range (10, 100): # genera i
valori che vanno da 10 a 99
    file.write("('Lav0000'+str(x)+'', 'Nome-Lavoratore0000'+str(x)+'', 'Cognome-
Lavoratore0000'+str(x)+''), \n")

for x in range (100, 1000): # genera i
valori che vanno da 100 a 999
    file.write("('Lav000'+str(x)+'', 'Nome-Lavoratore000'+str(x)+'', 'Cognome-
Lavoratore000'+str(x)+''), \n")

for x in range (1000, 10000): # genera i
valori che vanno da 1000 a 9999
    file.write("('Lav00'+str(x)+'', 'Nome-Lavoratore00'+str(x)+'', 'Cognome-
Lavoratore00'+str(x)+''), \n")

for x in range (10000, 100000): # genera i
valori che vanno da 10000 a 99999
    file.write("('Lav0'+str(x)+'', 'Nome-Lavoratore0'+str(x)+'', 'Cognome-
Lavoratore0'+str(x)+''), \n")

for x in range (100000, 200001): # genera i
valori che vanno da 100000 a 200000
    if x == 200000:
        file.write("('Lav'+str(x)+'', 'Nome-Lavoratore'+str(x)+'', 'Cognome-
Lavoratore'+str(x)+''); \n")
    else:
        file.write("('Lav'+str(x)+'', 'Nome-Lavoratore'+str(x)+'', 'Cognome-
Lavoratore'+str(x)+''), \n")

print("Scrittura Lavoratore terminata") # messaggio
a schermo
file.close() # chiude il
file

#-----#

file = open('MongoDB/File SQL 200k/Mandija_Cliente_200k.sql', 'w') # crea un f
ile TXT in scrittura

```

```

file.write('insert into clienteMongoDB values \n')          # scrivo ne
l file TXT la stringa

for x in range (1, 10):                                     # genera i
valori che vanno da 1 a 9
    file.write("('Cl100000"+str(x)+"', 'Nome-Cliente00000"+str(x)+"'), \n")

for x in range (10, 100):                                   # genera i
valori che vanno da 10 a 99
    file.write("('Cl10000"+str(x)+"', 'Nome-Cliente00000"+str(x)+"'), \n")

for x in range (100, 1000):                                  # genera i
valori che vanno da 100 a 999
    file.write("('Cl1000"+str(x)+"', 'Nome-Cliente0000"+str(x)+"'), \n")

for x in range (1000, 10000):                                # genera i
valori che vanno da 1000 a 9999
    file.write("('Cl100"+str(x)+"', 'Nome-Cliente000"+str(x)+"'), \n")

for x in range (10000, 100000):                              # genera i
valori che vanno da 10000 a 99999
    file.write("('Cl10"+str(x)+"', 'Nome-Cliente000"+str(x)+"'), \n")

for x in range (100000, 200001):                             # genera i
valori che vanno da 100000 a 200000
    if x == 200000:
        file.write("('Cl1"+str(x)+"', 'Nome-Cliente"+str(x)+"'); \n")
    else:
        file.write("('Cl1"+str(x)+"', 'Nome-Cliente"+str(x)+"'), \n")

print("Scrittura Cliente terminata")                         # messaggio
a schermo
file.close()                                                 # chiude il
file

#-----#

file = open('MongoDB/File SQL 200k/Mandija_Trasporto_200k.sql', 'w') # crea un f
ile TXT in scrittura
file.write('insert into trasportoMongoDB values \n')        # scrivo ne
l file TXT la stringa

for x in range (1, 10):                                       # genera i
valori che vanno da 1 a 9
    file.write("('Trp00000"+str(x)+"', 'Marca00000"+str(x)+"', 'Modello00000"+str(x)
)+'''), \n")

for x in range (10, 100):                                     # genera i
valori che vanno da 10 a 99

```

```

    file.write("('Trp0000'+str(x)+'", 'Marca0000'+str(x)+'", 'Modello0000'+str(x)+'"
'), \n")

for x in range (100, 1000):                                # genera i
valori che vanno da 100 a 999
    file.write("('Trp000'+str(x)+'", 'Marca000'+str(x)+'", 'Modello000'+str(x)+'"),
\ n")

for x in range (1000, 10000):                              # genera i
valori che vanno da 1000 a 9999
    file.write("('Trp00'+str(x)+'", 'Marca00'+str(x)+'", 'Modello00'+str(x)+'"), \n
")

for x in range (10000, 100000):                            # genera i
valori che vanno da 10000 a 99999
    file.write("('Trp0'+str(x)+'", 'Marca0'+str(x)+'", 'Modello0'+str(x)+'"), \n")

for x in range (100000, 200001):                          # genera i
valori che vanno da 100000 a 200000
    if x == 200000:
        file.write("('Trp'+str(x)+'", 'Marca'+str(x)+'", 'Modello'+str(x)+'"); \n")
    else:
        file.write("('Trp'+str(x)+'", 'Marca'+str(x)+'", 'Modello'+str(x)+'"), \n")

print("Scrittura Trasporto terminata")                    # messaggio
a schermo
file.close()                                              # chiude il
file

#-----
-----#

file = open('MongoDB/File SQL 200k/Mandija_Regione_200k.sql', 'w')    # crea un f
ile TXT in scrittura
file.write('insert into regioneMongoDB values \n')                # scrivo ne
l file TXT la stringa

for x in range (1, 10):                                        # genera i
valori che vanno da 1 a 9
    file.write("('Reg00000'+str(x)+'", 'Nome-Regione00000'+str(x)+'"), \n")

for x in range (10, 100):                                    # genera i
valori che vanno da 10 a 99
    file.write("('Reg0000'+str(x)+'", 'Nome-Regione0000'+str(x)+'"), \n")

for x in range (100, 1000):                                  # genera i
valori che vanno da 100 a 999
    file.write("('Reg000'+str(x)+'", 'Nome-Regione000'+str(x)+'"), \n")

for x in range (1000, 10000):                                # genera i
valori che vanno da 1000 a 9999

```

```

    file.write("('Reg00"+str(x)+"', 'Nome-Regione00"+str(x)+"'), \n")

for x in range (10000, 100000):                                # genera i
valori che vanno da 10000 a 99999
    file.write("('Reg0"+str(x)+"', 'Nome-Regione0"+str(x)+"'), \n")

for x in range (100000, 200001):                                # genera i
valori che vanno da 100000 a 200000
    if x == 200000:
        file.write("('Reg"+str(x)+"', 'Nome-Regione"+str(x)+"'); \n")
    else:
        file.write("('Reg"+str(x)+"', 'Nome-Regione"+str(x)+"'), \n")

print("Scrittura Regione terminata")                             # messaggio
a schermo
file.close()                                                     # chiude il
file

```

Inserimento valori generati casualmente

```

source "C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-
21/Dispense/Basi di Dati e di Conoscenza/Progetto/MongoDB/File SQL 200k/Mandija_Lav
oratore_200k.sql"
source "C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-
21/Dispense/Basi di Dati e di Conoscenza/Progetto/MongoDB/File SQL 200k/Mandija_Cli
ente_200k.sql"
source "C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-
21/Dispense/Basi di Dati e di Conoscenza/Progetto/MongoDB/File SQL 200k/Mandija_Tra
sporto_200k.sql"
source "C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-
21/Dispense/Basi di Dati e di Conoscenza/Progetto/MongoDB/File SQL 200k/Mandija_Reg
ione_200k.sql"
source "C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-
21/Dispense/Basi di Dati e di Conoscenza/Progetto/MongoDB/File SQL 200k/Mandija_Int
ervento_200k.sql"

```

Creazione viste tabelle

```
create view lavoratore as
select *
from lavoratoreMongoDB;

create view cliente as
select *
from clienteMongoDB;

create view trasporto as
select *
from trasportoMongoDB;

create view regione as
select *
from regioneMongoDB;

create view intervento as
select *
from interventoMongoDB;
```

Creazione collezioni JSON

```
select json_object("id_lavoratore", id_lavoratore, "nome_lavoratore", nome_lavoratore, "cognome_lavoratore", cognome_lavoratore)
into outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Mandija_Lavoratore_MongoDB.json'
from lavoratoreMongoDB;

select json_object("id_cliente", id_cliente, "nome_cliente", nome_cliente)
into outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Mandija_Cliente_MongoDB.json'
from clienteMongoDB;

select json_object("id_trasporto", id_trasporto, "marca", marca, "modello", modello)
into outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Mandija_Trasporto_MongoDB.json'
from trasportoMongoDB;

select json_object("id_regione", id_regione, "nome_regione", nome_regione)
into outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Mandija_Regione_MongoDB.json'
from regioneMongoDB;
```

```
select json_object("id_intervento", id_intervento, "id_lavoratore", id_lavoratore,
"id_cliente", id_cliente, "id_trasporto", id_trasporto, "id_regione", id_regione)
into outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Mandija_Intervento_MongoDB.json'
from interventoMongoDB;
```

Importazioni collezioni in MongoDB

```
mongoimport --db radio --collection lavoratore --drop --file
"C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-21/Dispense/Basi di Dati e di
Conoscenza/Progetto/MongoDB/File JSON/Mandija_lavoratore_MongoDB.json"
```

```
mongoimport --db radio --collection cliente --drop --file
"C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-21/Dispense/Basi di Dati e di
Conoscenza/Progetto/MongoDB/File JSON/Mandija_Cliente_MongoDB.json"
```

```
mongoimport --db radio --collection trasporto --drop --file
"C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-21/Dispense/Basi di Dati e di
Conoscenza/Progetto/MongoDB/File JSON/Mandija_Trasporto_MongoDB.json"
```

```
mongoimport --db radio --collection regione --drop --file
"C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-21/Dispense/Basi di Dati e di
Conoscenza/Progetto/MongoDB/File JSON/Mandija_Regione_MongoDB.json"
```

```
mongoimport --db radio --collection intervento --drop --file
"C:/Users/Thomas/Documents/Uni-TorVergata/Anno 20-21/Dispense/Basi di Dati e di
Conoscenza/Progetto/MongoDB/File JSON/Mandija_Intervento_MongoDB.json"
```

Esecuzione comandi MongoDB

```
show dbs
```

```
use radio
```

Query: MySQL vs MongoDB

1. visualizzare il cliente con il seguente nome: 056910

```
select cliente.nome_cliente
from cliente
where cliente.nome_cliente = 'Nome-Cliente056910';
```

```
db.cliente.find(
  {
    "nome_cliente": "Nome-Cliente056910"
  }
).explain("executionStats")
```

Nome-Cliente056910	}, "executionStats" : { "executionSuccess" : true, "nReturned" : 200, "executionTimeMillis" : 102,
Nome-Cliente056910	
+-----+ 200 rows in set (0.15 sec)	

```
2. visualizzare la regione con il seguente nome: 000050
```

```
select regione.nome_regione
from regione
where regione.nome_regione = 'Nome-Regione000050';
```

```
db.regione.find(
  {
    "nome_regione": "Nome-Regione000050"
  },
  {
    "nome_regione": 1
  }
).explain("executionStats")
```

Nome-Regione000050	
Nome-Regione000050	
Nome-Regione000050	
Nome-Regione000050	
+-----+	
51 rows in set (0.16 sec)	