

```
CREATE DATABASE dipPrj_p199;
```

```
USE dipPrj_p199;
```

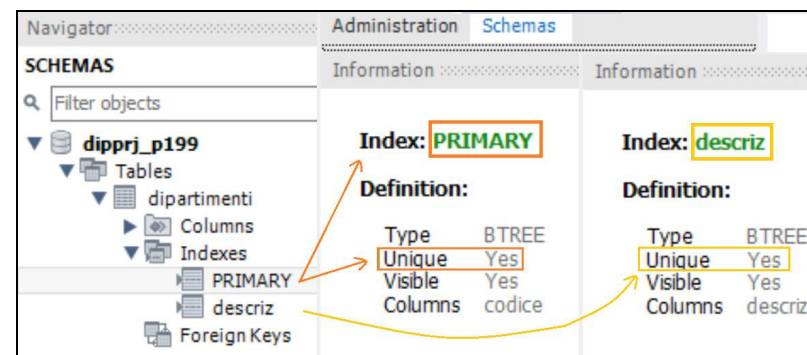
```
CREATE TABLE dipartimenti (
  codice          CHAR(5),
  descriz        VARCHAR(20)    NOT NULL,
  sede           VARCHAR(20),
  PRIMARY KEY    (codice),      -- index PRIMARY
  UNIQUE         (descriz)     -- index descriz
);
```

```
CREATE TABLE progetti (
  codPrj         CHAR(10),
  nomePrj        VARCHAR(40)    NOT NULL,
  durata         SMALLINT       NOT NULL,
  budget         DECIMAL(12)    NOT NULL,
  PRIMARY KEY    (codPrj)
);
```

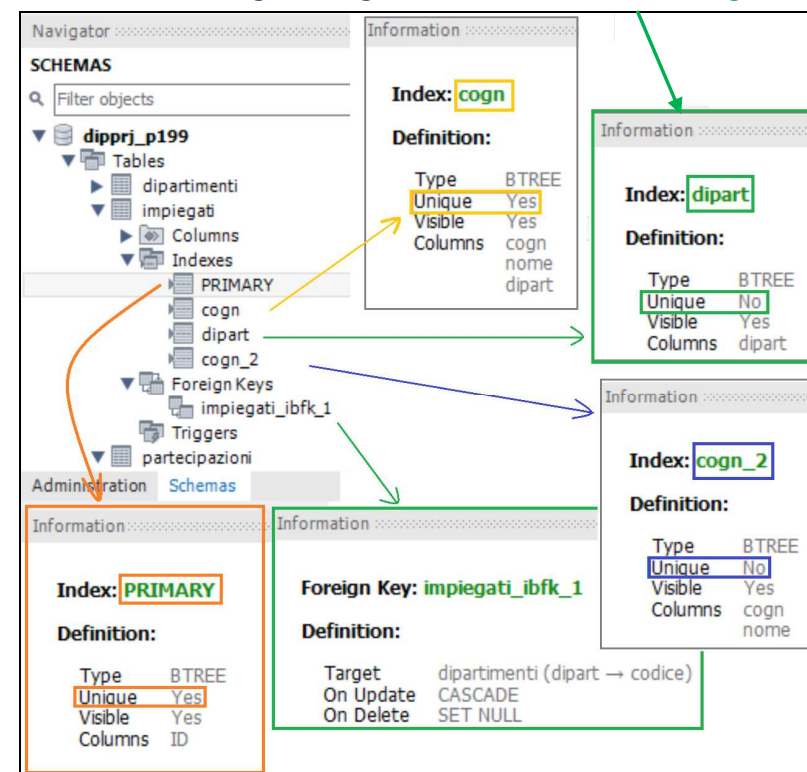
```
CREATE TABLE impiegati (
  ID              SMALLINT      AUTO_INCREMENT,
  nome            VARCHAR(20)    NOT NULL,
  cogn            VARCHAR(30)    NOT NULL,
  residenza      VARCHAR(20)    DEFAULT ("**manca residenza**"),
  stipendio      DECIMAL(9,2),
  dipart         CHAR(5),
  PRIMARY KEY    (ID),          -- index PRIMARY
  FOREIGN KEY    (dipart) REFERENCES dipartimenti (codice) -- in.dipart
  ON DELETE SET NULL          -- regole integrità referenziale in. impiegati_ibfk_1
  ON UPDATE CASCADE,         -- regole integrità referenziale in. impiegati_ibfk_1
  UNIQUE         (cogn, nome, dipart), -- index cogn
  INDEX         (cogn, nome)         -- index cogn_2
);
```

```
CREATE TABLE partecipazioni (
  ID              SMALLINT,
  codPrj          CHAR(10),
  ruolo           VARCHAR(30),
  percent         SMALLINT,
  PRIMARY KEY    (ID, codPrj), -- index PRIMARY
  FOREIGN KEY    (ID) REFERENCES impiegati (ID),
  FOREIGN KEY    (codPrj) REFERENCES progetti (codPrj) -- index codPrj
);
```

Oltre agli indici che scaturiscono dalle definizioni di PRIMARY KEY e FOREIGN KEY, si possono costruire indici aggiuntivi (esempio: l'indice univoco **descriz** sulla tabella **dipartimenti**), così esposti in MySQL



Naturalmente, per le **PRIMARY KEY** gli indici sono **UNIQUE**, mentre per le **FOREIGN KEY** gli indici generalmente **non sono UNIQUE**



Per la tabella **impiegati** MySQL crea 4 indici:

- **PRIMARY** sulla colonna ID
- **dipart** per la FOREIGN KEY sulla colonna **dipart** collegata a **dipartimenti.codice**
- **cogn** indice univoco sulle 3 colonne: **cogn**, **nome**, **dipart**
- **cogn_2** indice **non** univoco sulle 2 colonne: **cogn**, **nome**

il nome degli indici (in assenza di una indicazione esplicita) è determinato da MySQL in basa alla prima colonna indicata.

The screenshot shows the 'impiegati' table structure in the Schemas view. It includes columns for 'PRIMARY', 'dipart', 'cogn', and 'cogn_2'. A green arrow points from the 'dipart' column to a detailed view of the foreign key 'impiegati_ibfk_1'.

Foreign Key: impiegati_ibfk_1

Definition:

Target	dipartimenti (dipart → codice)
On Update	CASCADE
On Delete	SET NULL

Nella sezione **Foreign Key impiegati_ibfk_1** (nome assegnato da MySQL in assenza di indicazione) vengono riportati i **vincoli di integrità referenziale** (p.112) per **impiegati.dipart** :

- **ON DELETE SET NULL** : per cancellazione di una riga di **dipartimenti** tutte le righe collegate di **impiegati** (quelle con valore della colonna **impiegati.dipart = dipartimenti.codice** del dipartimento cancellato) vanno modificate impostando **impiegati.dipart** a NULL;
- **ON UPDATE CASCADE** : per modifica della colonna **codice** di **dipartimenti** tutte le righe collegate di **impiegati** vanno modificate impostando in **impiegati.dipart** il nuovo valore assegnato a **dipartimenti.codice**.

Per la tabella **partecipazioni** MySQL crea 2 indici:

- **PRIMARY** sulle colonne (**ID**, **codPrj**)
- **codPrj** per la FOREIGN KEY sulla colonna **codPrj** collegata a **progetti.codPrj**

The screenshot shows the 'partecipazioni' table structure in the Schemas view. It includes columns for 'PRIMARY', 'codPrj', and 'Foreign Keys'. Three detailed views are shown: the PRIMARY index, and two foreign keys: 'partecipazioni_ibfk_1' and 'partecipazioni_ibfk_2'. Green arrows point from the 'PRIMARY' and 'codPrj' index definitions to their respective foreign key definitions.

Index: PRIMARY

Definition:

Type	BTREE
Unique	Yes
Visible	Yes
Columns	ID, codPrj

Foreign Key: partecipazioni_ibfk_1

Definition:

Target	impiegati (ID → ID)
On Update	RESTRICT
On Delete	RESTRICT

Foreign Key: partecipazioni_ibfk_2

Definition:

Target	progetti (codPrj → codPrj)
On Update	RESTRICT
On Delete	RESTRICT

Nella sezione **Foreign Key** è indicato il vincolo **RESTRICT** per entrambe le FK (**partecipazioni_ibfk_1** per **ID** e **partecipazioni_ibfk_2** per **codPrj**).

RESTRICT è il vincolo di default che impedisce di cancellare le righe o modificare la PRIMARY KEY delle tabelle **parent** (**impiegati** o **progetti**) se ci sono ancora righe nella tabella **child** (**partecipazioni**) con FOREIGN KEY (**ID** o **codPrj**) con valore uguale a quello della riga che si sta cancellando o di cui si sta modificando la PRIMARY KEY.

L'opzione **NO ACTION** equivale alla opzione **RESTRICT**.

```
INSERT INTO dipartimenti VALUES
('Amm',      'Amministrazione', 'Roma'),
('Direz',    'Direzione Generale', 'Roma'),
('Mag',      'Magazzino',          'Torino'),
('Mkt',      'Marketing',          'Milano'),
('Pers',    'Personale',          'Roma'),
('Prod',    'Produzione',          'Torino'),
('R&S',    'Ricerca e Sviluppo', 'Torino');
```

```
INSERT INTO impiegati (cogn, nome, dipart, stipendio)
VALUES ('Rossi', 'Mario', 'Prod', 32000); -- ID sarà impostato a 1
```

```
INSERT INTO impiegati (ID, cogn, nome, residenza, stipendio)
VALUES ( 5, 'Marco', 'Viola', 'Palermo', 28300); -- dipart a NULL
```

```
INSERT INTO impiegati
VALUES ( 3, 'Mario', 'Rossi', 'Torino', 11500, 'Prod');
```

```
7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0
1 row(s) affected
1 row(s) affected
Error Code: 1062. Duplicate entry 'Rossi-Mario-Prod' for key 'impiegati.cogn'
```

ID	nome	cogn	residenza	stipendio	dipart
1	Mario	Rossi	**manca residenza**	32000.00	Prod
5	Viola	Marco	Palermo	28300.00	NULL

```
INSERT INTO impiegati
VALUES ( 3, 'Mario', 'Rossi', 'Torino', 11500, 'Mag'),
( 6, 'Enrico', 'Mori', 'Torino', 25000, 'Mag'),
(10, 'Margherita', 'Colombo', 'Torino', 65000, 'Prod');
```

```
UPDATE impiegati
SET stipendio = 21500
WHERE ID = 3;
```

```
UPDATE dipartimenti
SET codice = 'provamodif'
WHERE codice = 'prod';
```

```
SELECT * FROM impiegati; -- poi controaggiornare dipartimenti
```

```
INSERT INTO progetti VALUES
('Domo' , 'Componenti per domotica',          9, 150000),
('EleBike', 'Bici a pedalata assistita',      14, 300000),
('Metro' , 'Controllo traffico vetture metro', 18, 760000),
('Robot' , 'Controllo e attuatori robot industriali', 30, 1600000);
```

```
INSERT INTO partecipazioni VALUES
( 1, 'Domo' , 'Tecnico' , 40),
( 1, 'Metro' , 'Tecnico' , 50),
( 5, 'EleBike', 'Finanza' , 9),
( 5, 'Metro' , 'Finanza' , 10),
( 6, 'Domo' , 'Materiali', 12),
( 6, 'Metro' , 'Materiali', 30),
(10, 'Domo' , 'Direttore', 50);
```

```
INSERT INTO partecipazioni VALUES
(10, 'Domo' , 'Tecnico' , 40);
```

```
Error Code: 1062. Duplicate entry '10-Domo' for key 'partecipazioni.PRIMARY'
```

```
DELETE FROM progetti
WHERE codPrj = 'Domo';
```

```
Error Code: 1451. Cannot delete or update a parent row:
a foreign key constraint fails
(`dipprj_p199`.`partecipazioni`,
CONSTRAINT `partecipazioni_ibfk_2`
FOREIGN KEY (`codPrj`) REFERENCES `progetti` (`codPrj`))
```

```
UPDATE impiegati
SET ID = ID+1
WHERE ID = 6; -- OK per ID = 3 che non ha childs in partecipazioni
```

```
Error Code: 1451. Cannot delete or update a parent row:
a foreign key constraint fails
(`dipprj_p199`.`partecipazioni`,
CONSTRAINT `partecipazioni_ibfk_1`
FOREIGN KEY (`ID`) REFERENCES `impiegati` (`ID`))
```