SQL

DDL Fundamentals

Basic data types

- Numeric
  - smallint, integer, bigint, decimal, numeric, real, double precision, smallserial, serial, bigserial
- Character
  - character(n), char(n), varchar(n), text
- Date/time
  - timestamp, date, time, interval
- Boolean
  - boolean
- Geometric
  - …

	numero integer	nome character varying (20)
1	1	Cuccureddu
2	2	Perlina

```
create table prova_serial(
  numero serial, nome varchar(20));
insert into prova_serial(nome)
  values ('Cuccureddu'), ('Perlina');
select * from prova_serial;
```

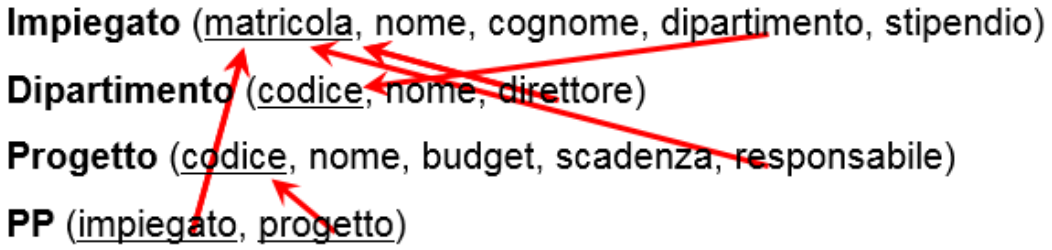
Basic data types

```sql
create table Impiegato(
  Matricola varchar(20),
  Nome varchar(50),
  Cognome varchar(50),
  Dipartimento varchar(50),
  Stipendio decimal(10,2));

create table Dipartimento(
  Codice varchar(50),
  Nome varchar(50),
  Direttore varchar(20));

create table Progetto(
  Codice varchar(50),
  Nome varchar(50),
  Budget decimal(10,2),
  Responsabile varchar(20));

create table PP(
  Impiegato varchar(50),
  Progetto varchar(50));
```



Basic constraints

- Check constraint

```
1  create table Impiegato(
2    Matricola varchar(20),
3    Nome varchar(50),
4    Cognome varchar(50),
5    Dipartimento varchar(50),
6    Stipendio decimal(10,2) check (stipendio>=0) );
7
8  create table Dipartimento(
9    Codice varchar(50),
10   Nome varchar(50),
11   Direttore varchar(20));
12
13 create table Progetto(
14   Codice varchar(50),
15   Nome varchar(50),
16   Budget decimal(10,2),
17   Responsabile varchar(20));
18
19 create table PP(
20   Impiegato varchar(50),
21   Progetto varchar(50));
```

Impiegato (matricola, nome, cognome, dipartimento, stipendio)

Dipartimento (codice, nome, direttore)

Progetto (codice, nome, budget, scadenza, responsabile)

PP (impiegato, progetto)

Basic constraints

- It is worth assigning a name to each constraint

```
1  create table Impiegato(
2    Matricola varchar(20),
3    Nome varchar(50),
4    Cognome varchar(50),
5    Dipartimento varchar(50),
6    Stipendio decimal(10,2) CONSTRAINT stipendio_positivo
7      check (stipendio>=0) );
8
9  create table Dipartimento(
10   Codice varchar(50),
11   Nome varchar(50),
12   Direttore varchar(20));
13
14 create table Progetto(
15   Codice varchar(50),
16   Nome varchar(50),
17   Budget decimal(10,2),
18   Responsabile varchar(20));
19
20 create table PP(
21   Impiegato varchar(50),
22   Progetto varchar(50));
```

Basic constraints

- Primary key constraint

```
1  create table Impiegato(
2    Matricola varchar(20) primary key,
3    Nome varchar(50),
4    Cognome varchar(50),
5    Dipartimento varchar(50),
6    Stipendio decimal(10,2) CONSTRAINT stipendio_positivo
7      check (stipendio>=0) );
8
9  create table Dipartimento(
10   Codice varchar(50) primary key,
11   Nome varchar(50),
12   Direttore varchar(20));
13
14 create table Progetto(
15   Codice varchar(50) primary key,
16   Nome varchar(50),
17   Budget decimal(10,2),
18   Responsabile varchar(20));
19
20 create table PP(
21   Impiegato varchar(50),
22   Progetto varchar(50),
23     primary key (impiegato, progetto) );
```

Basic constraints

- Not null constraint

```
1  create table Impiegato(
2    Matricola varchar(20) primary key,
3    Nome varchar(50) not null,
4    Cognome varchar(50) not null,
5    Dipartimento varchar(50),
6    Stipendio decimal(10,2) CONSTRAINT stipendio_positivo
7      check (stipendio>=0) );
8
9  create table Dipartimento(
10   Codice varchar(50) primary key,
11   Nome varchar(50) not null,
12   Direttore varchar(20));
13
14 create table Progetto(
15   Codice varchar(50) primary key,
16   Nome varchar(50) not null,
17   Budget decimal(10,2),
18   Responsabile varchar(20));
19
20 create table PP(
21   Impiegato varchar(50),
22   Progetto varchar(50),
23     primary key (impiegato, progetto) );
```

Basic constraints

- Unique constraint
  - Unique constraints ensure that the data contained in a column or a group of columns is unique with respect to all the rows in the table

- Notice that the primary key constraint can be assigned only to ONE group of columns:
  - In case of tables with multiple columns that can serve as primary key, one is defined as primary key and the others are defined as unique not null

Basic constraints

- Foreign key constraint

If no column name is specified in the references statement (just the table name) the primary key of the referenced table is used as the referenced column (lines 14, 20, 24, 26)

```
1  create table Impiegato(
2    Matricola varchar(20) primary key,
3    Nome varchar(50) not null,
4    Cognome varchar(50) not null,
5    Dipartimento varchar(50) CONSTRAINT dipartimento_fk
6      references Dipartimento (Codice),
7    Stipendio decimal(10,2) CONSTRAINT stipendio_positivo
8      check (stipendio>=0) );
9
10 create table Dipartimento(
11   Codice varchar(50) primary key,
12   Nome varchar(50) not null,
13   Direttore varchar(20) CONSTRAINT Direttore_fk
14     references Impiegato);
15
16 create table Progetto(
17   Codice varchar(50) primary key,
18   ) not null,
19   (10,2),
20   archar(20) references Impiegato);
21
22
23   Impiegato varchar(50),
24   Progetto varchar(50) references Progetto,
25     primary key (impiegato, progetto),
26     foreign key (impiegato) references Impiegato);
```

Impiegato (matricola, nome, cognome, dipartimento, stipendio)

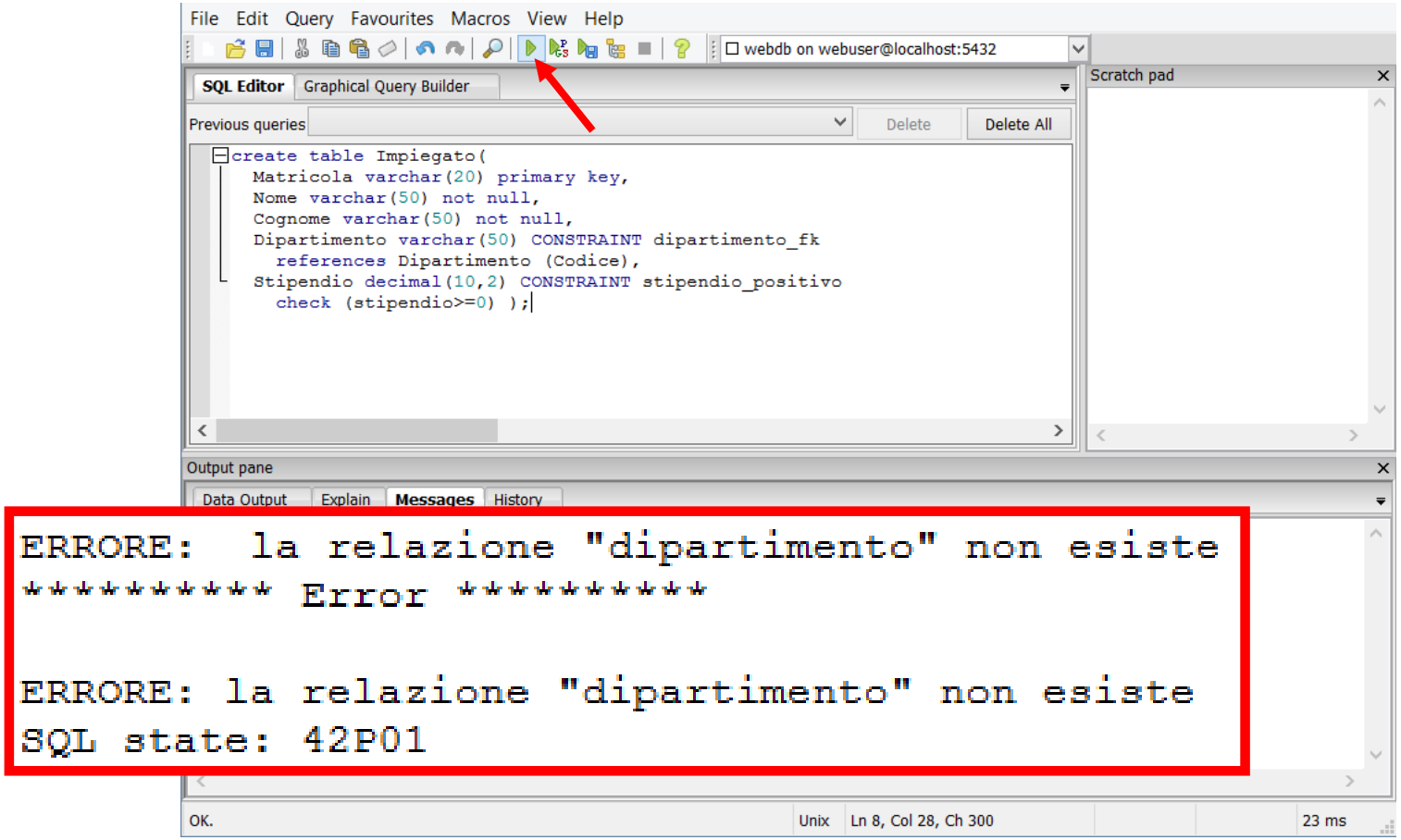
Dipartimento (codice, nome, direttore)

Progetto (codice, nome, budget, scadenza, responsabile)

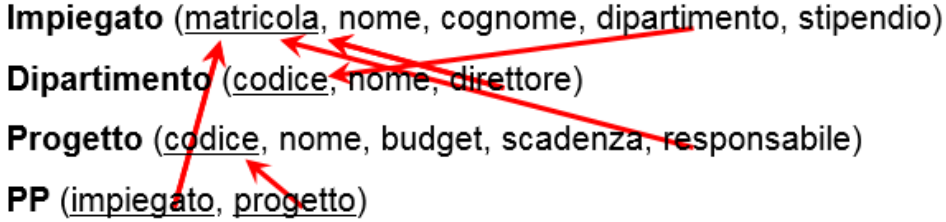
PP (impiegato, progetto)

Creating a table

- Copy and paste the first 8 lines into the SQL editor and execute the statement



Correct statements




```
1  create table Impiegato(
2    Matricola varchar(20) primary key,
3    Nome varchar(50) not null,
4    Cognome varchar(50) not null,
5    Dipartimento varchar(50),
6    Stipendio decimal(10,2) CONSTRAINT stipendio_positivo
7      check (stipendio>=0) );
8
9  create table Dipartimento(
10   Codice varchar(50) primary key,
11   Nome varchar(50) not null,
12   Direttore varchar(20) CONSTRAINT Direttore_fk
13     references Impiegato);
14
15 alter table Impiegato add constraint dipartimento_fk
16   foreign key (Dipartimento) references Dipartimento (Codice);
17
18 create table Progetto(
19   Codice varchar(50) primary key,
20   Nome varchar(50) not null,
21   Budget decimal(10,2),
22   Responsabile varchar(20) references Impiegato);
23
24 create table PP(
25   Impiegato varchar(50),
26   Progetto varchar(50) references Progetto,
27     primary key (impiegato, progetto),
28     foreign key (impiegato) references Impiegato);
```

Data insert

- To add a new row to a table, use the INSERT command
  - The command requires the table name and column values.
- Should we start adding rows to the table Impiegato or to the table Dipartimento?

```
insert into Impiegato values('AA998CC', 'Carlo', 'Cancelli', 'D001', 88000);
```

```
insert into Dipartimento values('D001', 'Amministrazione', 'AA998CC');
```

```
ERRORE:  la INSERT o l'UPDATE sulla tabella "impiegato" viola il vincolo di chiave esterna "dipartimento_fk"
DETAIL:  La chiave (dipartimento)=(D001) non è presente nella tabella "dipartimento".

********** Error **********

ERRORE: la INSERT o l'UPDATE sulla tabella "impiegato" viola il vincolo di chiave esterna "dipartimento_fk"
SQL state: 23503
Detail: La chiave (dipartimento)=(D001) non è presente nella tabella "dipartimento".
```

Data insert

```sql
insert into Dipartimento values('D001', 'Amministrazione', null);
insert into Dipartimento values('D002', 'Commerciale', null);
insert into Dipartimento values('D003', 'Direzione', null);
insert into Dipartimento values('D004', 'Ricerca', null);
insert into Dipartimento values('D005', 'Sviluppo', null);

insert into Impiegato values('AA998CC', 'Carlo', 'Cancelli', 'D001', 88000);
insert into Impiegato values('AA999BK', 'Giuliano', 'Casini', 'D002', 10500);
insert into Impiegato values('AA999BB', 'Francesco', 'Casini', 'D002', 104500);
insert into Impiegato values('AB999RT', 'Simone', 'Fochini', 'D001', 22000);
insert into Impiegato values('AD645GG', 'Serse', 'Moschini', 'D001', 66000);
insert into Impiegato values('AD999RT', 'Alfredo', 'Alfredi', 'D002', 71500);
insert into Impiegato values('AF145GG', 'Emilio', 'Sottani', 'D002', 71500);
insert into Impiegato values('AF444ED', 'Gino', 'Rampollo', 'D001', 79000);
insert into Impiegato values('AF445ED', 'Gino', 'Rampollo', 'D003', 89000);
insert into Impiegato values('AF676GR', 'Giuliano', 'Magri', 'D001', 93500);
insert into Impiegato values('AF978BF', 'Davide', 'Turli', null, 16500);
insert into Impiegato values('AG642GZ', 'Paolino', 'Galletto', 'D001', null);
insert into Impiegato values('AG978BF', 'Giulio', 'Magri', 'D003', 44000);
insert into Impiegato values('AH756RF', 'Carla', 'Bruni', 'D004', 77000);
insert into Impiegato values('AS764TK', 'Angela', 'Gogoli', 'D001', 38500);
insert into Impiegato values('BF132FR', 'Augusto', 'Meda', 'D002', 41800);
insert into Impiegato values('BH756RF', 'Luca', 'Frondi', 'D005', 60500);
insert into Impiegato values('RF132FR', 'Laura', 'Casini', 'D004', 99000);
insert into Impiegato values('RF133FR', 'Laura', 'Casini', 'D005', 100000);

update Dipartimento set direttore = 'AA998CC' where codice= 'D001';
update Dipartimento set direttore = 'AD999RT' where codice= 'D002';
update Dipartimento set direttore = 'AF978BF' where codice= 'D003';
update Dipartimento set direttore = 'AH756RF' where codice= 'D004';
update Dipartimento set direttore = 'RF133FR' where codice= 'D005';

insert into Progetto values('P001', 'Reti in fibra', 300000.00, 'AA998CC');
```

Cascading DML actions with FK

- If an INSERT, UPDATE, DELETE statement violates an intra-relational constraint, the DMBS reports an error and the statement is not executed

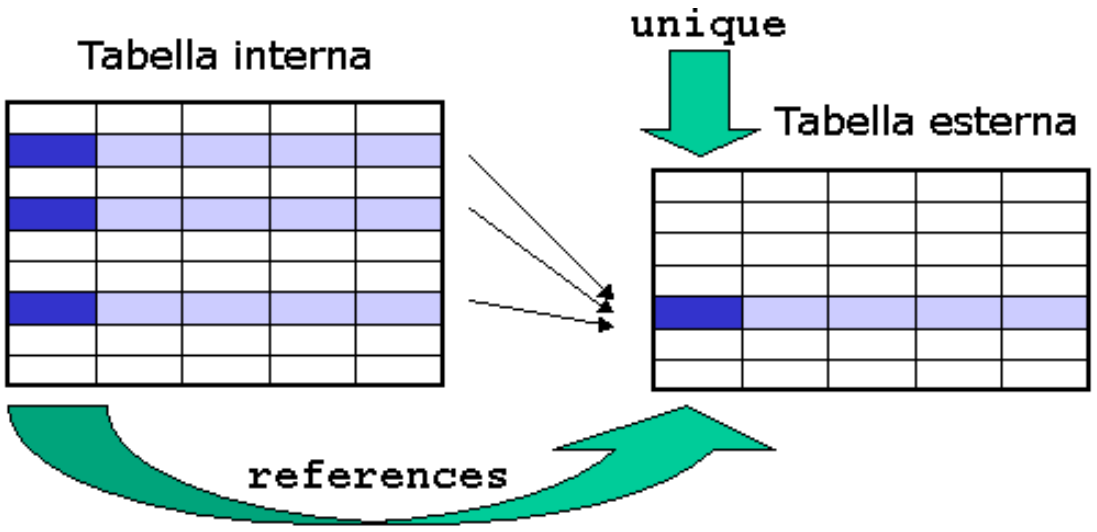
- If an INSERT, UPDATE, DELETE statement violates a foreign key constraint, different actions can take place

Cascading DML actions with FK

- When a foreign key constraint is created, it is possible to specify some compensating action that is executed in case the FK constraint is violated

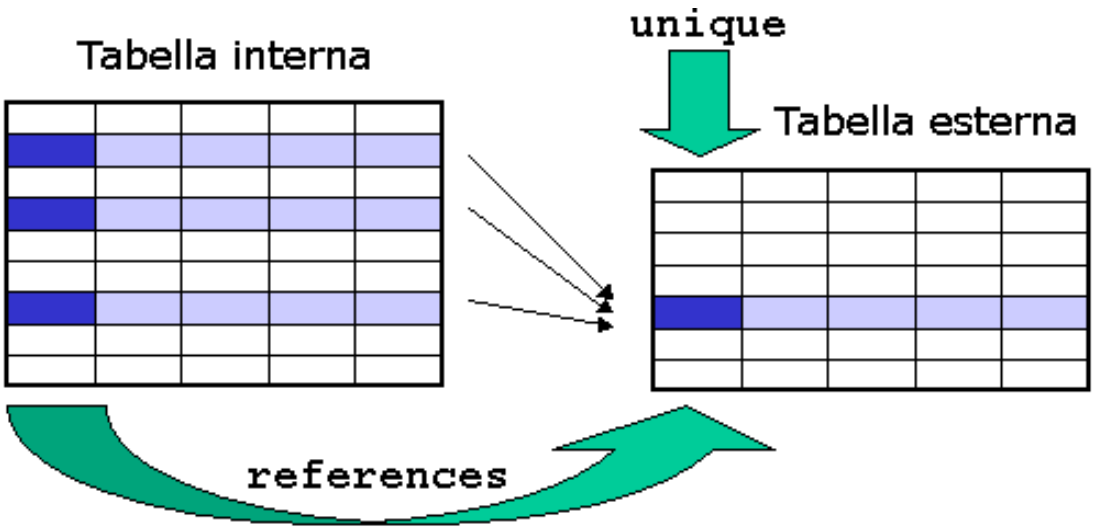
Cascading DML actions with FK

- The FK contraint identifies one internal and one external table
- The constraint can be violated by a statement that acts on the internal table, on the external one or on both



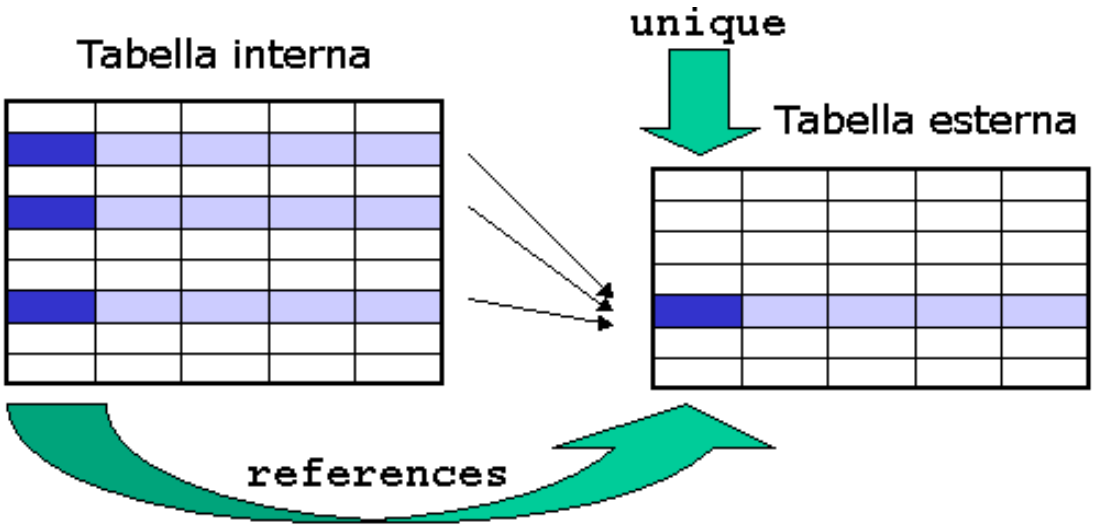
Cascading DML actions with FK

- **Violation** by statements acting on the **internal table**:
  - INSERT or UPDATE a record with a value (in the FK constrained attributes) that doesn't match any value in the referenced attributes
  - In both cases the DBMS aborts the statement and returns an error



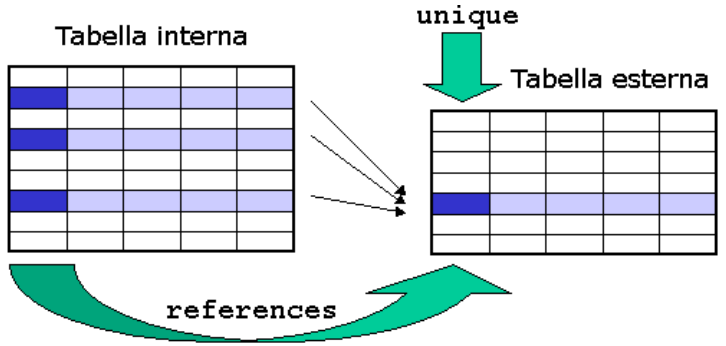
Cascading DML actions with FK

- **Violation** by statements acting on the **external table**:
  - DELETE or UPDATE a record with a value (in the FK constrained attributes) that is referenced by some rows of the internal table
  - In both cases, it is possible to specify some instructions that tell the DBMS how to compensate for the violation



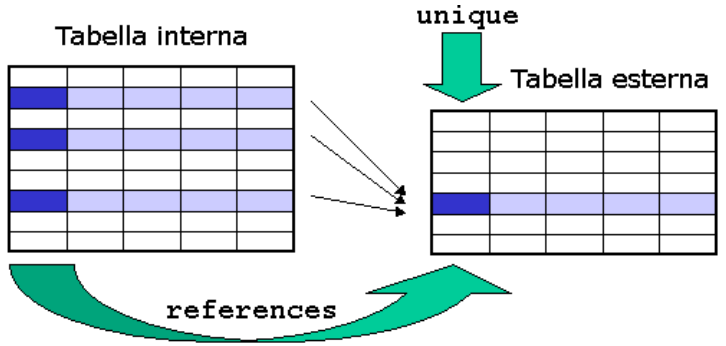
Cascading DML actions with FK

- On DELETE
  - `cascade`: all the records in the internal table that reference the deleted record are deleted
  - `set null`: all the records in the internal table that reference the deleted record are updated so that the referencing attribute is set to NULL
  - `set default`: all the records in the internal table that reference the deleted record are updated so that the referencing attribute is set to the DEFAULT value
  - `noaction`: an error is reported and the DELETE statement is not executed



Cascading DML actions with FK

- On UPDATE
  - `cascade`: all the records in the internal table that reference the updated record are updated accordingly
  - `set null`: all the records in the internal table that reference the updated record are updated so that the referencing attribute is set to NULL
  - `set default`: all the records in the internal table that reference the updated record are updated so that the referencing attribute is set to the DEFAULT value
  - `noaction`: an error is reported and the UPDATE statement is not executed



Cascading DML actions with FK

- It is possible to specify different compensating actions to manage FK violations by DELETE and UPDATE statements:

```
alter table Impiegato add constraint dipartimento_fk
    foreign key (Dipartimento) references
    Dipartimento (Codice) ON DELETE set null
    ON UPDATE cascade;
```