

Tabella delle entità

Nome	Descrizione
Users	Dati relativi agli
Plans	Dati relativi ai piani promozionali
Keys	Dati relativi alle chiavi

Tabella degli attributi dell'entità Utenti

Entità	Nome	Tipo	Dimensione	Vincoli	Descrizione
Users	IDUser	int	6	Chiave primaria in auto-incremento	Identificatore univoco degli utenti
	name	varchar	30	//	Nome
	surname	varchar	30	//	Cognome
	email	varchar	64	//	E-mail
	password	varchar	60	//	Password

Tabella degli attributi dell'entità Plans

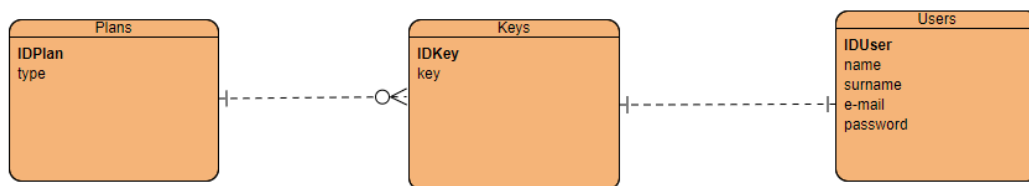
Entità	Nome	Tipo	Dimensione	Vincoli	Descrizione
Plans	IDPlan	int	1	Chiave primaria in auto-incremento	Identificatore univoco dei piani
	type	varchar	16	//	Tipo di piano

Tabella degli attributi dell'entità Keys

Entità	Nome	Tipo	Dimensione	Vincoli	Descrizione
Keys	IDKey	int	6	Chiave primaria in auto-incremento	Identificatore univoco delle chiavi
	expiration_date	datetime	//	> now	Data di scadenza della chiave
	key	varchar	36	//	Nome della città

Tabella delle relazioni

Nome	Entità 1	Entità 2	Cardinalità	Descrizione
UsersKeys	Users	Keys	1-1	Un utente può avere una sola chiave, una chiave può essere posseduta da un solo utente.
KeysPlans	Plans	Keys	1 - M	Una chiave ha un solo piano, un piano può essere posseduto da più chiavi



## Schema logico

Plans (IDPlan, type)

Users (IDUser, name, surname, e-mail, password)

Keys(IDKey, key, expiration\_date, CODPlan\*, CODUser\*)

```

create table Users(
IDUser int(6) auto_increment,
name varchar (30) not null,
surname varchar (30) not null,
email varchar (64) not null unique,
password varchar (60) not null,
primary key(IDUser));

```

```

create table Keys(
IDKey int(6) auto_increment,
key varchar(36) not null unique,
expiration_date datetime,
CODPlan int(6),
CODUser int(6),
primary key(IDKey),
FOREIGN KEY (CODPlan) REFERENCES Plans(IDPlan),
FOREIGN KEY (CODUser) REFERENCES Plans(IDUser));

```

```

create table Plans(

```

```
IDPlan int(1) auto_increment,  
type varchar(16) not null);
```