

Appello di Basi di Dati - 21/07/2016
Proff. Tiziana Catarci e Andrea Marrella
A.A. 2015-2016

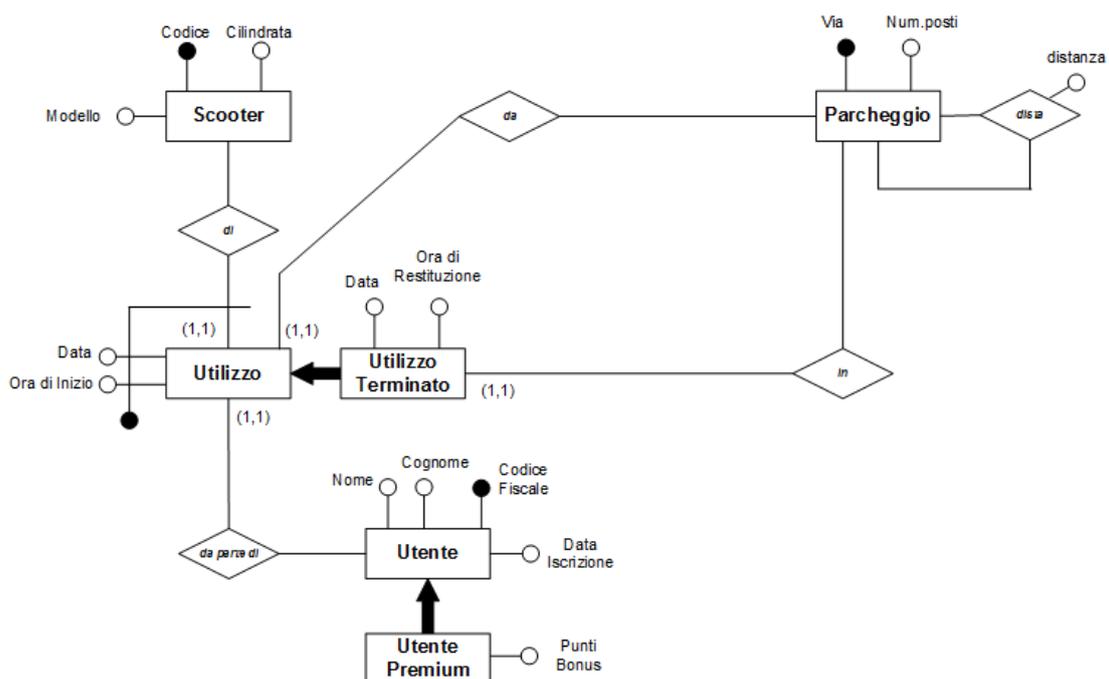
DOMANDA 1 (10 punti)

Si vuole progettare lo schema ER di una base di dati che contiene le informazioni relative alla gestione dello scooter sharing (condivisione di scooter) organizzato da un comune.

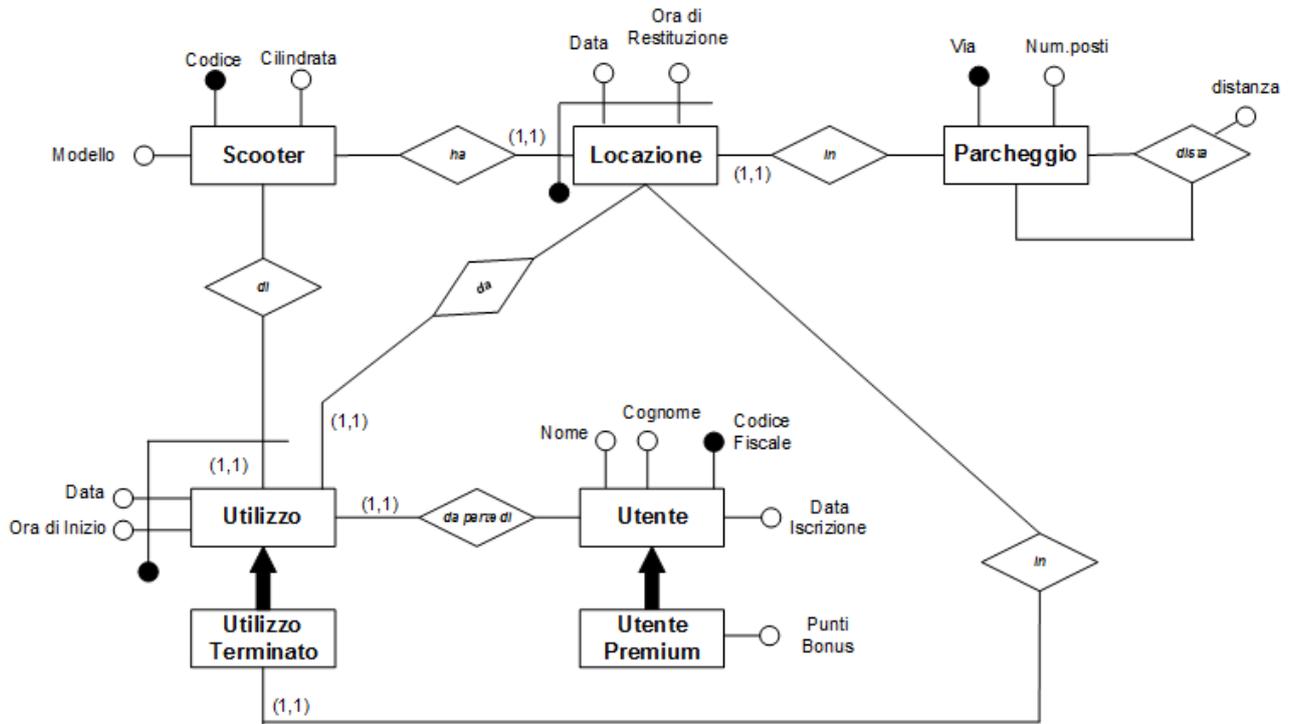
Di uno scooter interessa il codice (identificativo), il modello e la cilindrata. Gli scooter vengono prelevati e restituiti presso degli appositi parcheggi situati in città. Nel caso in cui lo scooter sia stato restituito, è di interesse memorizzare il parcheggio nel quale è avvenuta la restituzione con data e ora di restituzione. Di un parcheggio di scooter interessa la via in cui è situato ed il numero di posti di cui dispone. Si noti che non ci possono essere due parcheggi nella stessa via. Inoltre, interessa conoscere quanto dista (in metri) un parcheggio da ogni altro parcheggio di scooter della città. Di un utilizzo di scooter da parte di un utente del servizio interessa conoscere l'ora e la data di inizio, lo scooter utilizzato (uno ed uno solo), l'utente ed il parcheggio da cui lo scooter è stato prelevato. Si noti che uno scooter non può essere utilizzato contemporaneamente da più utenti.

Di un utente generico del servizio di scooter sharing interessano il nome, il cognome, il codice fiscale (identificativo) e la data in cui si è iscritto al servizio. Di un utente premium interessa conoscere anche i punti bonus accumulati da quando si è iscritto.

SOLUZIONE 1



SOLUZIONE ALTERNATIVA

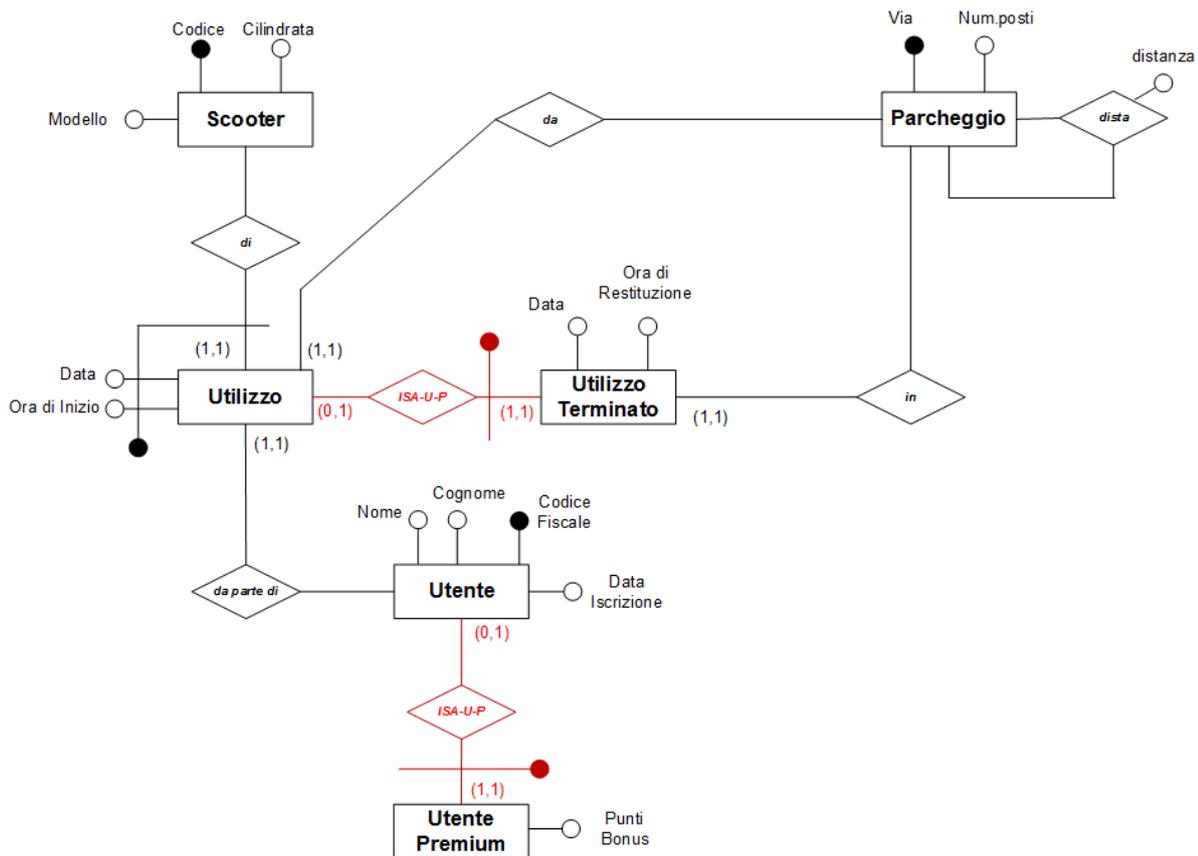


VINCOLO E STERNO: Per ogni istanza *u* di «Utilizzo», seguendo la relazione «*da*» e la relazione «*ha*» si ottiene un'istanza «*s*» di «Scooter» che si otterrebbe anche seguendo la relazione «*di*» da *u*.

DOMANDA 2 (8 punti)

Si effettui la progettazione logica del diagramma ER realizzato con riferimento alla domanda 1, seguendo l'indicazione di evitare il più possibile valori nulli nella base di dati.

SCHEMA RISTRUTTURATO (relativo a SOLUZIONE 1)



SCHEMA LOGICO (relativo a SOLUZIONE 1)

Scooter (Codice, Cilindrata, Modello)

Parcheggio(Via, NumPosti)

dista(Parcheggio1, Parcheggio2, distanza)

- *foreign key*:
dista[Parcheggio1] \subseteq Parcheggio[Codice]
- *foreign key*:
dista[Parcheggio2] \subseteq Parcheggio[Codice]

Utente(CodiceFiscale, Nome, Cognome, DataIscrizione)

UtentePremium(CodiceFiscale, PuntiBonus)

- *foreign key*:
UtentePremium [CodiceFiscale] \subseteq Utente[CodiceFiscale]

Utilizzo (Scooter, DataInizio, OralInizio, Utente, Parcheggio)

- *foreign key*:
Utilizzo [Scooter] \subseteq Scooter[Codice]
- *foreign key*:
Utilizzo [Utente] \subseteq Utente[CodiceFiscale]
- *foreign key*:
Utilizzo [Parcheggio] \subseteq Parcheggio[Via]

UtilizzoTerminato(Scooter, DataInizio, OralInizio, DataFine, OraFine, ParcheggioDiRestituzione)

- *foreign key*:
UtilizzoTerminato[Scooter, DataInizio, OralInizio] \subseteq Utilizzo[Scooter, DataInizio, OralInizio]
- *foreign key*:
UtilizzoTerminato[ParcheggioDiRestituzione] \subseteq Parcheggio[Via]

DOMANDA 3

Sia dato il seguente schema relazionale:

Paese (nome, nro-abitanti, provincia)

Percorso (partenza, arrivo, lunghezza)

in cui sia il campo *partenza* che il campo *arrivo* della tabella Percorso referenziano il campo *nome* della tabella Paese. Le lunghezze dei percorsi sono espressi in Km.

In riferimento a tale schema relazionale si esprimano le seguenti interrogazioni SQL:

1. Calcolare la lunghezza massima dei percorsi tra paesi con più di 100.000 abitanti (**3 punti**)

```
SELECT max(lunghezza)
FROM Percorso r, Paese p1, Paese p2
WHERE r.partenza = p1.nome AND r.arrivo = p2.nome AND
p1.nro_abitanti>100000 AND p2.nro_abitanti>100000
```

2. Calcolare per ogni provincia la lunghezza media del percorso fra i paesi di quella provincia (**4 punti**)

```
SELECT p1.provincia, AVG(lunghezza)
FROM Percorso r, Paese p1, Paese p2
WHERE r.partenza = p1.nome AND r.arrivo = p2.nome AND
p1.provincia = p2.provincia AND p1.nome <> p2.nome
GROUP BY p1.provincia
```

3. Identificare quei paesi per cui non esiste alcun paese che disti da loro meno di 100 Km (**4 punti**)

```
SELECT nome
FROM paese
EXCEPT
(SELECT p.partenza
FROM percorso p)
```

```
WHERE p.lunghezza < 100
UNION
SELECT p.arrivo
FROM percorso p
WHERE p.lunghezza < 100)
```