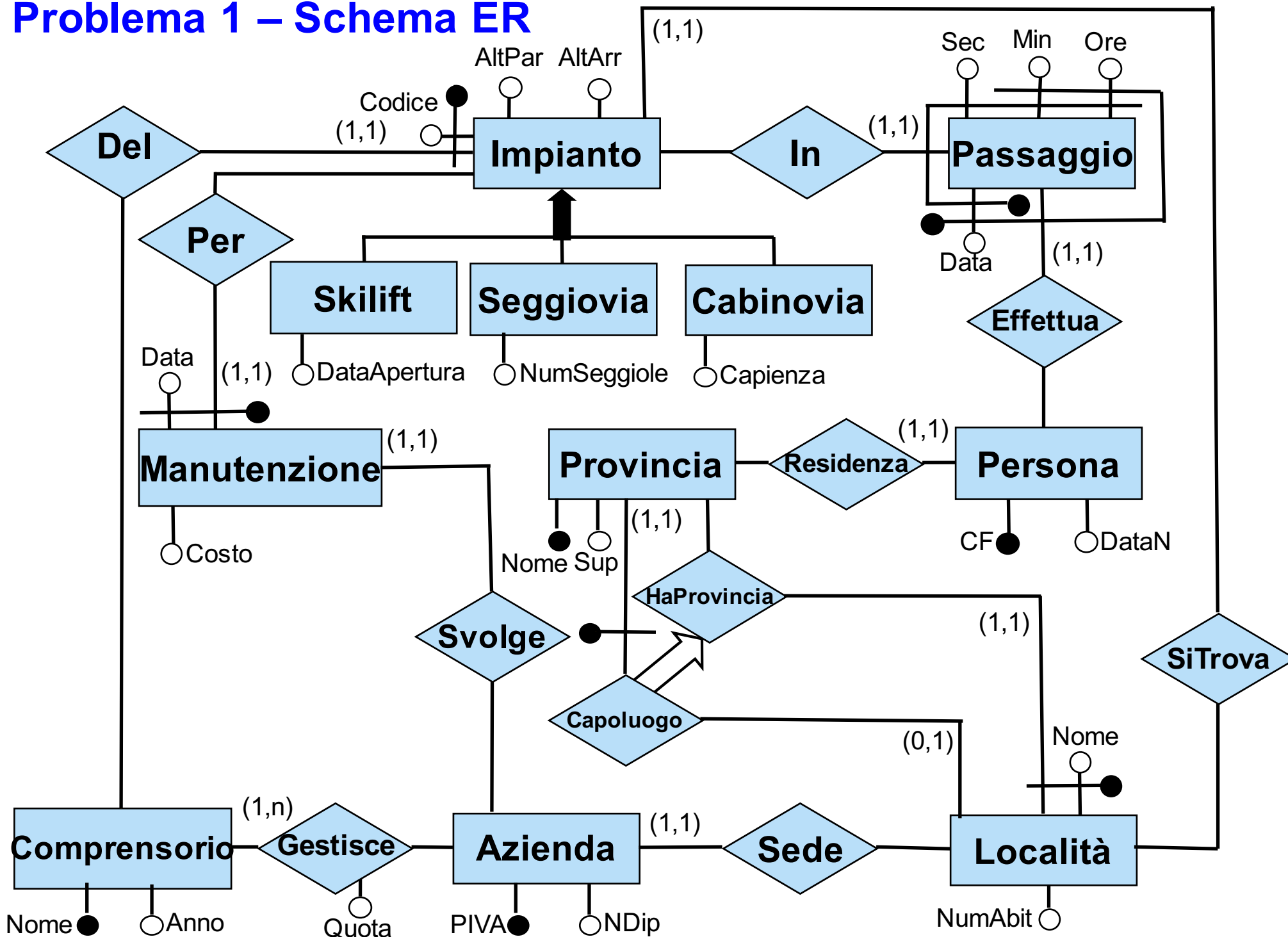


Basi di dati

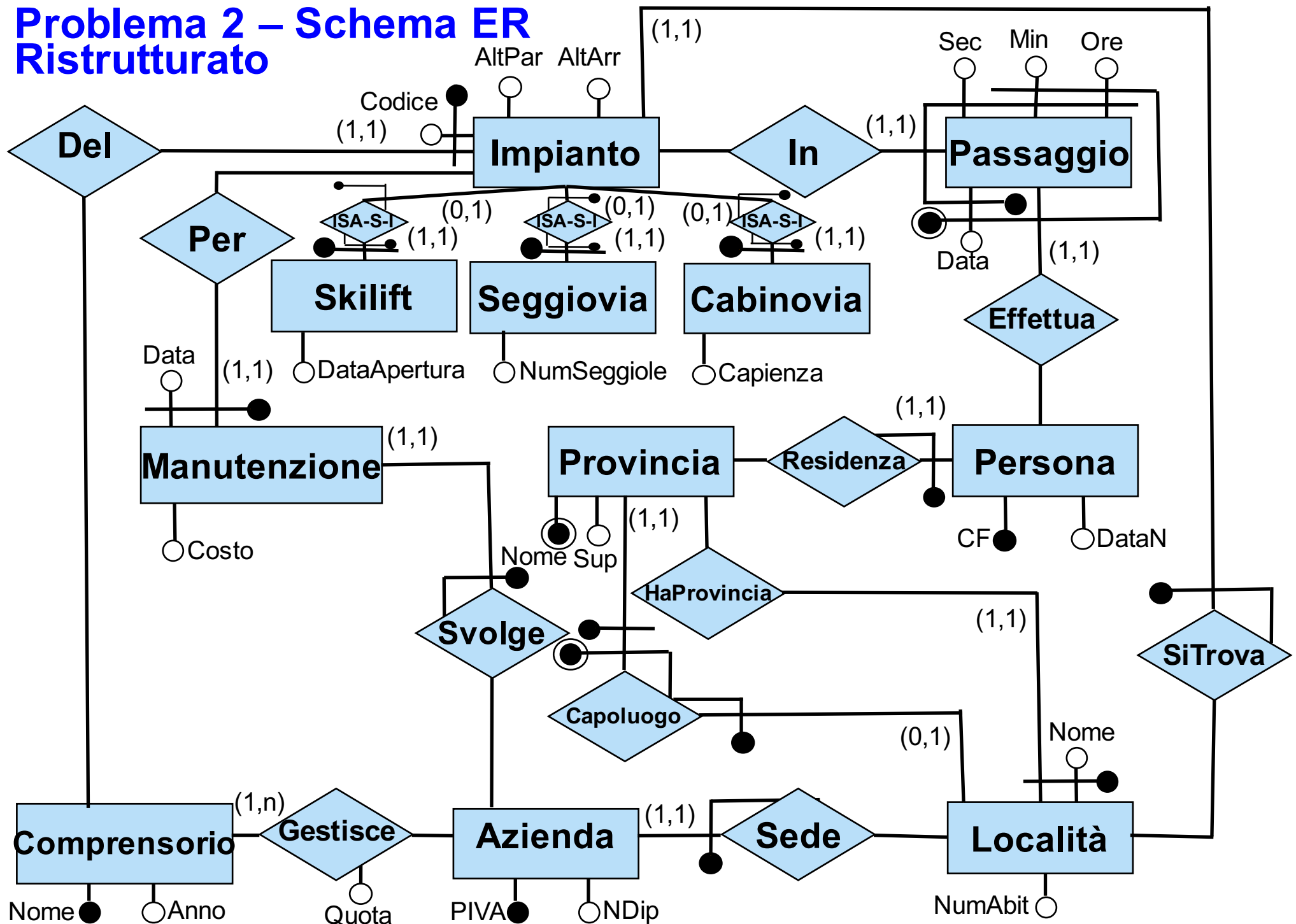
Appello del 25-01-2017
Compito A

Anno Accademico 2016/17

Problema 1 – Schema ER



Problema 2 – Schema ER Ristrutturato



Vincoli esterni: 1. vincoli di generalizzazione completa; 2. ogni istanza di Capoluogo è istanza di HaProvincia

Problema 2 - Schema logico dopo la traduzione diretta

Impianto(Codice, Compensorio, AltPar, AltArr)

foreign key: Impianto[Compensorio] \subseteq Compensorio[Codice]

vincolo di generalizzazione: Impianto[Codice,Compensorio] \subseteq Skilift[Codice,Compensorio] \cup
Seggiovia[Codice,Compensorio] \cup
Cabinovia[Codice,Compensorio]

Compensorio(Nome, Anno)

inclusione: Compensorio[Nome] \subseteq Gestisce[Compensorio]

Skilift(Codice, Compensorio, DataApertura)

foreign key: Skilift[Codice,Compensorio] \subseteq Impianto[Codice,Compensorio]

vincolo di disgiuntezza: Skilift[Codice,Compensorio] \cap Seggiovia[Codice,Compensorio] = \emptyset

vincolo di disgiuntezza: Skilift[Codice,Compensorio] \cap Cabinovia[Codice,Compensorio] = \emptyset

Seggiovia(Codice, Compensorio, NumSeggiole)

foreign key: Seggiovia[Codice,Compensorio] \subseteq Impianto[Codice,Compensorio]

vincolo di disgiuntezza: Seggiovia[Codice,Compensorio] \cap Cabinovia[Codice,Compensorio] = \emptyset

Cabinovia(Codice, Compensorio, Capienza)

foreign key: Cabinovia[Codice,Compensorio] \subseteq Impianto[Codice,Compensorio]

SiTrova(Codice, Compensorio, NomeLoc, ProvLoc)

foreign key: SiTrova[Codice,Compensorio] \subseteq Impianto[Codice,Compensorio]

foreign key: SiTrova[NomeLoc,ProvLoc] \subseteq Località[Nome,Provincia]

Località(Nome, Provincia, NumAbit)

foreign key: Località[Provincia] \subseteq Provincia[Nome]

Provincia(Nome, Sup)

foreign key: Provincia[Nome] \subseteq Capoluogo[Provincia]

Capoluogo(Provincia, NomeLoc, ProvLoc)

foreign key: Capoluogo[Provincia] \subseteq Provincia[Nome]

foreign key: Capoluogo[NomeLoc,ProvLoc] \subseteq Località[Nome,Provincia]

chiave: ProvLoc

vincolo di tupla: Provincia = ProvLoc

Azienda(PIVA, Ndip)

foreign key: Azienda[PIVA] \subseteq Sede[Azienda]

Problema 2 - Schema logico dopo la traduzione diretta

Sede(Azienda, NomeLoc, ProvLoc)

foreign key: Sede[Azienda] \subseteq Azienda[PIVA]

foreign key: Sede[NomeLoc,ProvLoc] \subseteq Località[Nome,Provincia]

Gestisce(Azienda, Compensorio, Quota)

foreign key: Gestisce[Azienda] \subseteq Azienda[PIVA]

foreign key: Gestisce[Compensorio] \subseteq Compensorio[Nome]

Manutenzione(Data, CodImp, CompImp, Costo)

foreign key: Manutenzione[CodImp,CompImp] \subseteq Impianto[Codice,Compensorio]

foreign key: Manutenzione[Data, CodImp, CompImp] \subseteq Svolge[DataMan,CodImpMan, CompImpMan]

Svolge(DataMan, CodImpMan, CompImpMan, Azienda)

foreign key: Svolge[DataMan,CodImpMan, CompImpMan] \subseteq Manutenzione[Data,CodImp,CompImp]

foreign key: Svolge[Azienda] \subseteq Azienda[PIVA]

Persona(CF, DataN)

foreign key: Persona[CF] \subseteq Residenza[Persona]

Residenza(Persona, Provincia)

foreign key: Residenza[Persona] \subseteq Persona[CF]

foreign key: Residenza[Provincia] \subseteq Provincia[Nome]

Passaggio(Persona, Data, Ore, Min, Sec)

foreign key: Passaggio[Persona] \subseteq Persona[CF]

In(PersPass, DataPass, OrePass, MinPass, CodImp, CompImp)

foreign key: In[PersPass,DataPass,OrePass,MinPass] \subseteq Passaggio[Persona,Data,Ore,Min]

foreign key: In[CodImp,CompImp] \subseteq Impianto[Codice,Compensorio]

vincolo esterno: nella relazione che è il risultato dell'equijoin tra In e Passaggio con condizione (PersPass=Persona and DataPass=Data and OrePass=Ore and MinPass=Min), gli attributi CodImp, CompImp, Data, Ore, Min, Sec formano una chiave

Problema 2 - Schema logico dopo la ristrutturazione

- *Indicazione 1*: Accorpamento di Manutenzione e Svolge (sfruttando l'accoppiamento forte)
- *Indicazione 2*: Accorpamento di Impianto con Skilift, Seggiovia, Cabinovia, con relativa introduzione dell'attributo "Tipo" (i cui valori possibili sono esattamente tre, 1 corrispondente a Skilift, 2 a Seggiovia e 3 a Cabinovia), e con indicazioni di possibili valori nulli negli attributi DataApertura, NumSeggiole e Capienza; successiva decomposizione per ricreare le relazioni Skilift e Cabinovia, riscorporandole da Impianto, al fine di evitare valori nulli.
- *Ulteriore ristrutturazione*: Accorpamento per eliminare l'attributo inutile ProvLoc in Capoluogo.

Impianto(Codice, Comprensorio, AltPar, AltArr, Tipo, NumSeggiole*)

foreign key: Impianto[Comprensorio] \subseteq Comprensorio[Codice]

vincolo di dominio: Tipo=1 OR Tipo=2 OR Tipo=3

vincolo di tupla: Tipo=1 SE E SOLO SE NumSeggiole \neq NULL

vincolo esterno: PROJ_{Codice,Comprensorio}SEL_{Tipo=1}(Impianto) \subseteq Skylift[Codice,Comprensorio]

vincolo esterno: PROJ_{Codice,Comprensorio}SEL_{Tipo=3}(Impianto) \subseteq Cabinovia[Codice,Comprensorio]

Comprensorio(Nome, Anno)

inclusione: Comprensorio[Nome] \subseteq Gestisce[Comprensorio]

Skilift(Codice, Comprensorio, DataApertura)

vincolo esterno: Skilift[Codice,Comprensorio] \subseteq PROJ_{Codice,Comprensorio}SEL_{Tipo=1}(Impianto)

Cabinovia(Codice, Comprensorio, Capienza)

vincolo esterno: Cabinovia[Codice,Comprensorio] \subseteq PROJ_{Codice,Comprensorio}SEL_{Tipo=3}(Impianto)

SiTrova(Codice, Comprensorio, NomeLoc, ProvLoc)

foreign key: SiTrova[Codice,Comprensorio] \subseteq Impianto[Codice,Coprensorio]

foreign key: SiTrova[NomeLoc,ProvLoc] \subseteq Località[Nome,Provincia]

Località(Nome, Provincia, NumAbit)

foreign key: Località[Provincia] \subseteq Provincia[Nome]

Provincia(Nome, Sup)

foreign key: Provincia[Nome] \subseteq Capoluogo[Provincia]

Capoluogo(Provincia, NomeLoc)

foreign key: Capoluogo[Provincia] \subseteq Provincia[Nome]

foreign key: Capoluogo[NomeLoc,ProvLoc] \subseteq Località[Nome,Provincia]

Problema 2 - Schema logico dopo la ristrutturazione

Azienda(PIVA,Ndip)

foreign key: Azienda[PIVA] ⊆ Sede[Azienda]

Sede(Azienda, NomeLoc, ProvLoc)

foreign key: Sede[Azienda] ⊆ Azienda[PIVA]

foreign key: Sede[NomeLoc,ProvLoc] ⊆ Località[Nome,Provincia]

Gestisce(Azienda, Compensorio, Quota)

foreign key: Gestisce[Azienda] ⊆ Azienda[PIVA]

foreign key: Gestisce[Compensorio] ⊆ Compensorio[Nome]

Manutenzione(Data, CodImp, CompImp, Costo, Azienda)

foreign key: Manutenzione[CodImp,CompImp] ⊆ Impianto[Codice,Compensorio]

foreign key: Manutenzione[Data, CodImp, CompImp] ⊆ Svolge[DataMan,CodImpMan, CompImpMan]

foreign key: Manutenzione[Azienda] ⊆ Azienda[PIVA]

Persona(CF, DataN)

foreign key: Persona[CF] ⊆ Residenza[Persona]

Residenza(Persona,Provincia)

foreign key: Residenza[Persona] ⊆ Persona[CF]

foreign key: Persona[CF] ⊆ Residenza[Persona]

Passaggio(Persona, Data, Ore, Min, Sec]

foreign key: Passaggio[Persona] ⊆ Persona[CF]

In(PersPass, DataPass, OrePass, MinPass,CodImp,CompImp)

foreign key: In[PersPass,DataPass,OrePass,MinPass] ⊆ Passaggio[Persona,Data,Ore,Min]

foreign key: In[CodImp,CompImp] ⊆ Impianto[Codice,Compensorio]

vincolo esterno: nella relazione che è il risultato dell'equijoin tra In e Passaggio con condizione (PersPass=Persona and DataPass=Data and OrePass=Ore and MinPass=Min), gli attributi CodImp, CompImp, Data, Ore, Min e Sec formano una chiave

Problema 3: soluzione

1. La risposta è positiva: esiste una base di dati T1 tale che la valutazione della query 1 su T1 dà un risultato non vuoto uguale alla valutazione della query 2 su T1. Infatti, basta considerare come T1 la base di dati che ha le seguenti tuple:

$\langle a1, b1 \rangle$ in R, $\langle b1, c1 \rangle$ in P, $\langle d1, e1 \rangle$ in Q.

È facile verificare che la valutazione di entrambe le query sulla base di dati T1 dà come risultato: $\{ \langle a1, c1 \rangle \}$.

1. La risposta è positiva: esiste una base di dati T2 tale che la valutazione della query 1 su T2 dà un risultato diverso dalla valutazione della query 2 su T2. Infatti, basta considerare come T2 la base di dati che ha le seguenti tuple:

$\langle a2, b2 \rangle$ in R, $\langle b2, c2 \rangle$ in P, $\langle c2, e2 \rangle$ in Q, $\langle d2, e3 \rangle$ in Q.

È facile verificare che la valutazione della query Q1 su T2 dà come risultato l'insieme vuoto mentre la valutazione della query Q2 su T2 dà come risultato: $\{ \langle a2, c2 \rangle \}$.

Problema 4: soluzione

Si omettono le definizioni di vincolo di integrità e di vincolo di cardinalità.

Non esiste alcuna istanza dello schema S in cui l'insieme delle istanze della entità F non è vuoto. Infatti, qualunque sia l'istanza I di S , se z_1 è in $Istanze(I, F)$, allora z_1 deve partecipare alla relazione Q nel ruolo F , e quindi in $Istanze(I, Q)$ ci sarà una tupla $\langle E:e, F:z_1 \rangle$, con e in $Istanze(I, G)$. Ma essendo e istanza di G , essa deve partecipare 3 volte alla relazione Q , e quindi in $Istanze(I, Q)$ ci saranno altre due tuple $\langle E:e, F:z_2 \rangle$ e $\langle E:e, F:z_3 \rangle$. Ma tutte le tuple che sono istanze di Q sono anche istanze di R , e quindi abbiamo che le tre tuple $\langle E:e, F:z_1 \rangle$, $\langle E:e, F:z_2 \rangle$ e $\langle E:e, F:z_3 \rangle$ sono tutte in $Istanze(I, R)$, e quindi violano il vincolo di cardinalità $(1, 2)$ che impone che ogni istanza di E partecipi al massimo due volte alla relazione R nel ruolo E .