

Esame di Basi di Dati

A.A. 2020/2021 – Appello del 29/03/2021

Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione relativo alle attività svolte dalle commissioni parlamentari nei vari anni. Di ogni commissione interessa il nome (identificativo), l'anno della sua istituzione ed il tipo (permanente, consultiva, ecc.). Ogni commissione è composta da almeno 5 parlamentari e se al primo giorno di ogni mese la composizione risulta cambiata, l'informazione di quali membri fanno parte della commissione deve essere registrata nella base di dati, insieme all'informazione di chi tra i membri svolge il ruolo di presidente e a quale gruppo parlamentare in quel momento tale presidente fa capo. Ogni commissione afferisce ad uno dei due rami del Parlamento, Camera o Senato. Delle commissioni della Camera interessa anche il codice dell'aula di Montecitorio assegnata per i suoi lavori e delle commissioni del Senato interessano le varie riunioni svolte, ciascuna con data di svolgimento e categoria (interlocutoria, plenaria, ecc.), tenendo presente che non si possono svolgere più riunioni di una commissione al giorno. Di ogni parlamentare interessa il codice fiscale (identificativo), il nome, il cognome, la data di nascita ed il partito politico al quale apparteneva al momento della sua entrata in Parlamento.

Problema 2

Si richiede di effettuare la progettazione logica partendo dallo schema concettuale prodotto per il problema 1, tenendo conto dell'indicazione che, data una commissione ed un mese di un anno, si vuole spesso conoscere il presidente della commissione in quel mese.

Problema 3

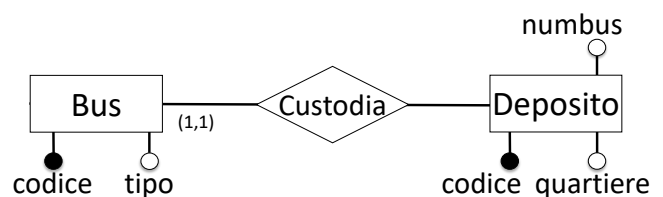
Si consideri una base di dati che comprende la relazione `Docente(codice, cittanascita, datanascita)` e la relazione `Concorso(docente, anno, punteggio)`, dove la prima memorizza il codice (chiave primaria), la città di nascita e la data di nascita di un insieme di docenti, mentre la seconda memorizza il codice del docente, l'anno ed il punteggio ottenuto in un concorso universitario al quale il docente ha partecipato (vige la regola che nessun docente può partecipare ai concorsi universitari più volte in un anno). Sappiamo che quando il punteggio di un concorso per un docente non è noto, nell'attributo `punteggio` per tale concorso è memorizzato il valore nullo, mentre negli altri attributi delle due relazioni non compaiono mai valori nulli. Sappiamo anche che è definito un vincolo di foreign key da `docente` a `codice`. Si scrivano in SQL le seguenti query: (1) Calcolare il codice di *tutti* i docenti, ognuno con l'indicazione di quanti sono i concorsi ai quali ha partecipato. (2) Per *ogni* docente nato a Roma, calcolare il codice, calcolare il punteggio minimo conseguito negli eventuali concorsi ai quali ha partecipato. (3) Dei docenti che hanno partecipato ad almeno 5 concorsi, calcolare il codice, la data di nascita ed il punteggio medio ottenuto nei concorsi nei quali il punteggio è noto.

Problema 4

Riferendosi alla base di dati del problema 3, si scrivano in algebra relazionale le seguenti query: (1) Calcolare le coppie di codici dei docenti che sono nati nella stessa città e che hanno partecipato ad almeno un concorso. (2) Calcolare il codice dei docenti che in tutti i concorsi a cui hanno partecipato hanno ottenuto un punteggio maggiore di 25.

Problema 5

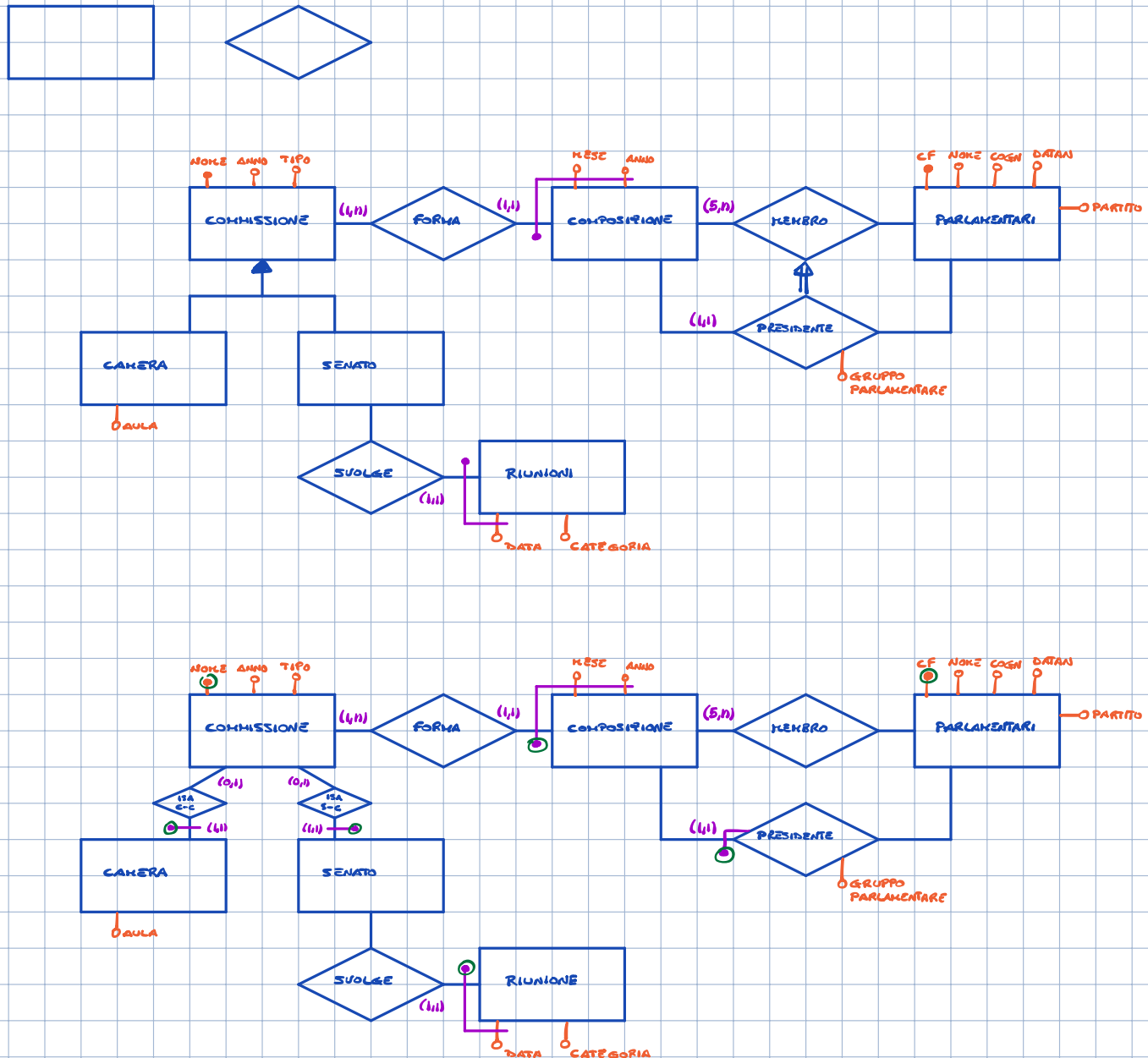
Nello schema concettuale S mostrato qui a destra, tutti gli attributi tranne `numbus` sono di tipo "varchar" e l'attributo `numbus` (di tipo intero) indica, per ogni istanza di `Deposito`, il numero di bus custoditi in quel deposito (come specificato dal vincolo esterno). Si sa a priori che non si eseguiranno cancellazioni di depositi, che quando si accede ad un bus si vuole sempre sapere il suo deposito di custodia e quando si accede ad un deposito si vuole sempre sapere il numero di bus che custodisce. Si chiede di eseguire la progettazione logica partendo dallo schema concettuale S esprimendo lo schema relazionale risultante in SQL.



Vincolo esterno: per ogni istanza I di S , per ogni d in $Istanze(I, Deposito)$, il valore n tale che $\langle d, n \rangle$ è in $Istanze(I, numbus)$ è pari al numero di istanze di Bus collegate a d dalla relazione $Custodia$.

Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione relativo alle attività svolte dalle commissioni parlamentari nei vari anni. Di ogni commissione interessa il nome (identificativo), l'anno della sua istituzione ed il tipo (permanente, consultiva, ecc.). Ogni commissione è composta da almeno 5 parlamentari e se al primo giorno di ogni mese la composizione risulta cambiata, l'informazione di quali membri fanno parte della commissione deve essere registrata nella base di dati, insieme all'informazione di chi tra i membri svolge il ruolo di presidente e a quale gruppo parlamentare in quel momento tale presidente fa capo. Ogni commissione afferisce ad uno dei due rami del Parlamento, Camera o Senato. Delle commissioni della Camera interessa anche il codice dell'aula di Montecitorio assegnata per i suoi lavori e delle commissioni del Senato interessano le varie riunioni svolte, ciascuna con data di svolgimento e categoria (interlocutoria, plenaria, ecc.), tenendo presente che non si possono svolgere più riunioni di una commissione al giorno. Di ogni parlamentare interessa il codice fiscale (identificativo), il nome, il cognome, la data di nascita ed il partito al quale apparteneva al momento della sua entrata in Parlamento.



- Ogni istanza di Presidente è anche istanza di Membro.
- Ogni istanza di Commissione partecipa a Camera, oppure a Senato, ma non ad entrambi.

Commissione (nome, anno, tipo)

inc Commissione(nome) \subseteq Commissione(nome)

vg Commissione(nome) \subseteq Camera(commissione) \cup Senato(commissione)

Camera(commissione, anno)

fu Camera(comm) \subseteq Commissione(nome)

vg Camera(commissione) \cap Senato(commissione) $= \emptyset$

Senato(commissione)

fu Senato(comm) \subseteq Commissione(nome)

Riunione (data, senato, categoria)

fu Riunione(senato) \subseteq Senato(commissione)

Commissione(commissione, meze, anno)

fu Commissione(commissione) \subseteq Commissione(nome)

inc Commissione(comm, meze, anno) \subseteq Commissione(comm, meze, anno)

fu Commissione(comm, meze, anno) \subseteq Presidente(comm, meze, anno)

Parlamentari(CF, nome, cogn, data, partito)

Membro(commissione, meze, anno, parlamentare)

fu Membro(comm, meze, anno) \subseteq Commissione(comm, meze, anno)

fu Membro(parlamentare) \subseteq Parlamentare(CF)

Presidente(commissione, meze, anno, parlamentare, gruppo)

fu Presidente(commissione, meze, anno) \subseteq Commissione(comm, meze, anno)

fu Presidente(parlamentare) \subseteq Parlamentare(CF)

Problema 3

Si consideri una base di dati che comprende la relazione `Docente(codice, cittanascita, datanascita)` e la relazione `Concorso(docente, anno, punteggio)`, dove la prima memorizza il codice (chiave primaria), la città di nascita e la data di nascita di un insieme di docenti, mentre la seconda memorizza il codice del docente, l'anno ed il punteggio ottenuto in un concorso universitario al quale il docente ha partecipato (vige la regola che nessun docente può partecipare ai concorsi universitari più volte in un anno). Sappiamo che quando il punteggio di un concorso per un docente non è noto, nell'attributo `punteggio` per tale concorso è memorizzato il valore nullo, mentre negli altri attributi delle due relazioni non compaiono mai valori nulli. Sappiamo anche che è definito un vincolo di foreign key da `docente` a `codice`. Si scrivano in SQL le seguenti query: (1) Calcolare il codice di *tutti* i docenti, ognuno con l'indicazione di quanti sono i concorsi ai quali ha partecipato. (2) Per *ogni* docente nato a Roma, calcolare il codice, calcolare il punteggio minimo conseguito negli eventuali concorsi ai quali ha partecipato. (3) Dei docenti che hanno partecipato ad almeno 5 concorsi, calcolare il codice, la data di nascita ed il punteggio medio ottenuto nei concorsi nei quali il punteggio è noto.

```
1) SELECT D.CODICE, COUNT(C.ANNO)
FROM DOCENTE D LEFT JOIN CONCORSO C ON D.CODICE = C.DOCENTE
GROUP BY D.CODICE
```

```
2) SELECT D.CODICE, MIN(C.PUNTEGGIO)
FROM DOCENTE D LEFT JOIN CONCORSO C ON D.CODICE = C.DOCENTE
WHERE D.CITTANASCITA = "ROMA"
GROUP BY D.CODICE
```

```
3) SELECT D.CODICE, D.DATANASCITA, AVG(C.PUNTEGGIO)
FROM DOCENTE D JOIN CONCORSO C ON D.CODICE = C.DOCENTE
GROUP BY D.CODICE, D.DATANASCITA
HAVING COUNT(*) >= 5
```

Problema 4

Riferendosi alla base di dati del problema 3, si scrivano in algebra relazionale le seguenti query: (1) Calcolare le coppie di codici dei docenti che sono nati nella stessa città e che hanno partecipato ad almeno un concorso. (2) Calcolare il codice dei docenti che in tutti i concorsi a cui hanno partecipato hanno ottenuto un punteggio maggiore di 25.

Si consideri una base di dati che comprende la relazione **Docente**(codice,cittanascita,datanascita) e la relazione **Concorso**(docente,anno,punteggio), dove la prima memorizza il codice (chiave primaria),

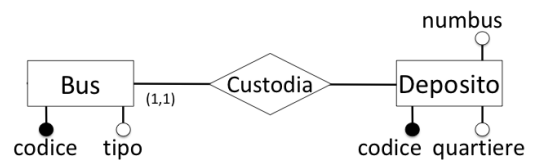
1)
$$\text{PROJ}_{\text{CODICE}, C} (\text{DOCENTE JOIN CONCONSO}) \text{ JOIN } [\text{REN}_{\text{CODICE} \neq C, \text{CITTANASCITA} \neq N, \dots} (\text{DOCENTE JOIN CONCONSO})]$$

AND
 $\text{CITTANASCITA} = N$

2)
$$\text{PROJ}_{\text{DOCENTE}} (\text{CONCONSO}) - \text{PROJ}_{\text{DOCENTE}} [\text{SEL}_{\text{PUNTEGGIO} \leq 25} (\text{CONCONSO})]$$

Problema 5

Nello schema concettuale S mostrato qui a destra, tutti gli attributi tranne **numbus** sono di tipo "varchar" e l'attributo **numbus** (di tipo intero) indica, per ogni istanza di **Deposito**, il numero di bus custoditi in quel deposito (come specificato dal vincolo esterno). Si sa a priori che non si eseguiranno cancellazioni di depositi, che quando si accede ad un bus si vuole sempre sapere il suo deposito di custodia e quando si accede ad un deposito si vuole sempre sapere il numero di bus che custodisce. Si chiede di eseguire la progettazione logica partendo dallo schema concettuale S esprimendo lo schema relazionale risultante in SQL.



Vincolo esterno: per ogni istanza l di S , per ogni d in $\text{Istanze}(l, \text{Deposito})$, il valore n tale che $\langle d, n \rangle$ è in $\text{Istanze}(l, \text{numbus})$ è pari al numero di istanze di Bus collegate a d dalla relazione **Custodia**.

Bus(codice, tipo)

$f_u \text{ Bus}[\text{codice}] \subseteq \text{Custodia}[\text{Bus}]$

Deposito(codice, quartiere, numbus)

Custodia(Bus, deposito)

$f_u \text{ Custodia}[\text{Bus}] \subseteq \text{Bus}[\text{codice}]$

$f_u \text{ Custodia}[\text{deposito}] \subseteq \text{Deposito}[\text{codice}]$

Accorpamento Bus e Custodia:

Bus(codice, tipo, deposito)

$f_u \text{ Bus}[\text{deposito}] \subseteq \text{Deposito}[\text{codice}]$

Deposito(codice, quartiere, numbus)

```

CREATE TABLE BUS {
    codice varchar(20),
    tipo varchar(20),
    deposito varchar(20),
    primary key codice,
    foreign key (deposito) reference Deposito(codice);
};

```

```

CREATE TABLE Deposito {
    codice varchar,
    quartiere varchar,
    numero integer,
    primary key codice;
};

```

```

CREATE OR REPLACE FUNCTION aggiorna_bus()

```

```

RETURNS TRIGGER AS

```

```

$$

```

```

BEGIN

```

```

    UPDATE DEPOSITO

```

```

    SET DEPOSITO.NUMBUS = (SELECT COUNT(*) FROM BUS WHERE DEPOSITO.CODICE = BUS.DEPOSITO);

```

```

END;

```

```

$$ PLSQL

```

```

CREATE TRIGGER assegna_numbus AFTER INSERT ON BUS
FOR EACH ROW EXECUTE PROCEDURE aggiorna_bus();

```