

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

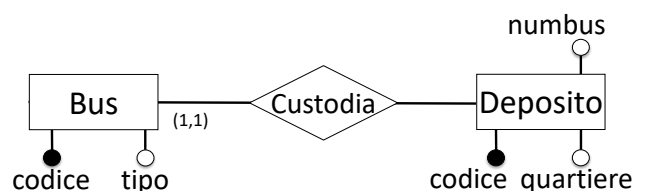
Si consideri una base di dati che comprende la relazione `Docente(codice, cittanascita, datanascita)` e la relazione `Concorso(docente, anno, punteggio)`, dove la prima memorizza il codice (chiave primaria), la città di nascita e la data di nascita di un insieme di docenti, mentre la seconda memorizza 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.

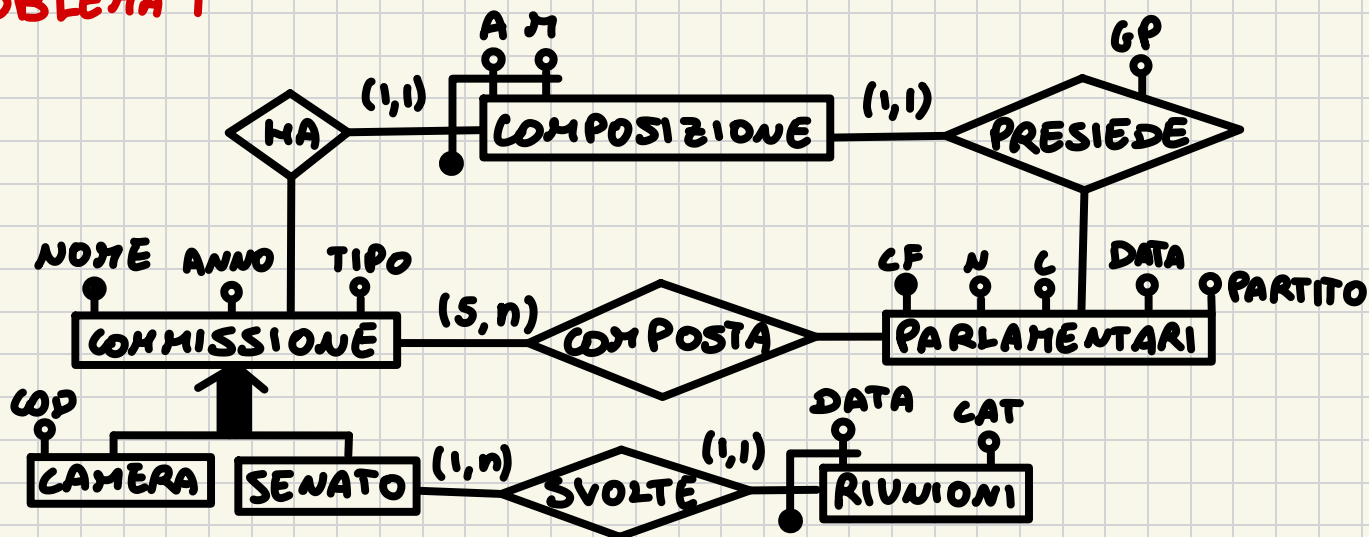
Problema 5

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.

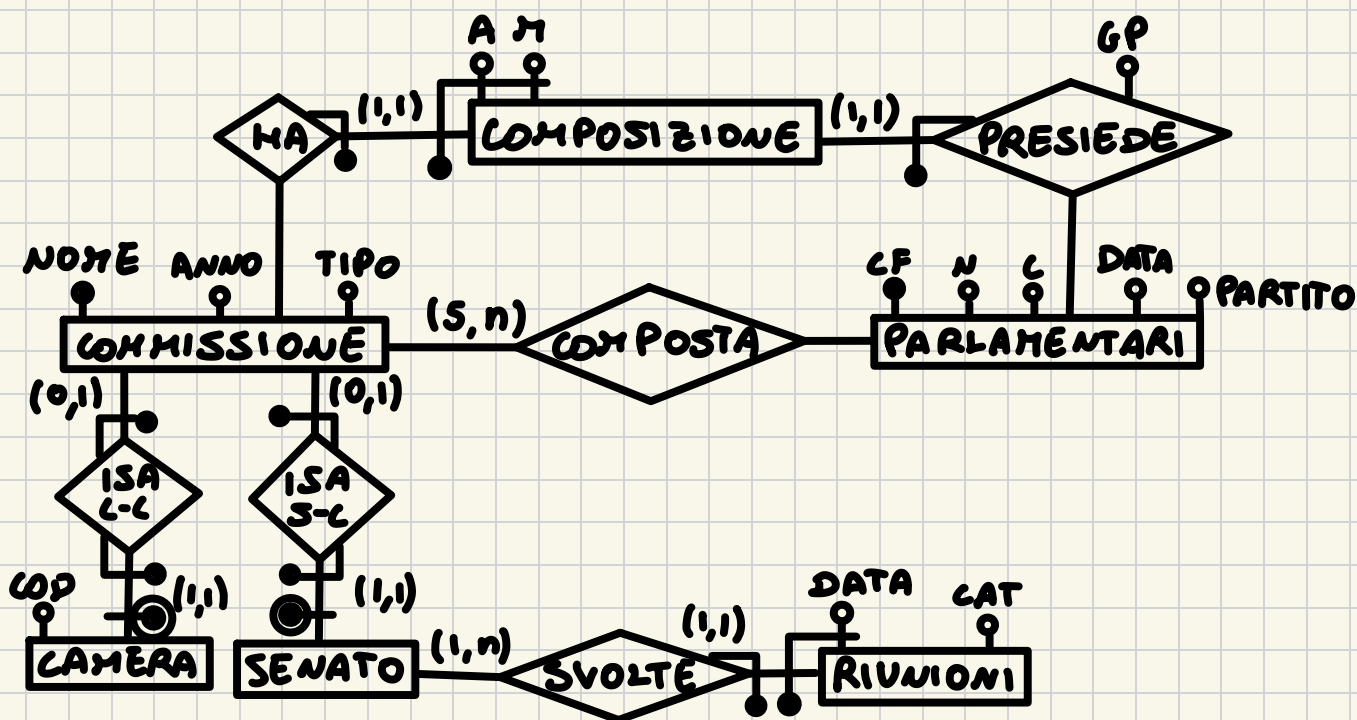


Vincolo esterno: per ogni istanza l di S , per ogni d in $Istanze(l, Deposito)$, il valore n tale che $\langle d, n \rangle$ è in $Istanze(l, numbus)$ è pari al numero di istanze di **Bus** collegate a d dalla relazione **Custodia**.

PROBLEMA 1



PROBLEMA 2



OGNI ISTANZA DI COMMISSIONE PARTECIPA AD ISA-C-C O AD ISA-S-C, MA NON ENTRAMBE

COMMISSIONE (NOME, ANNO, TIPO)

INCLUSIONE: COM[NOME] \subseteq COMP[COM]

PARLAMENTARI (CF, NOME, COGNOME, DATA, PARTITO)

COMPOSTA (COMMISSIONE, PARLAMENTARI)

FK: COM[COM] \subseteq COM[NOME]

FK: COM[PARL] \subseteq PARL[CF]

CAMERA (COM, COD)

FK: CAT[COM] \subseteq COM[COD]

SENATO (COM)

FK: SEN [COM] \subseteq COM [COD]

INCLUSIONE: SEN [COM] \subseteq RIUNIONE[SEN]

VINCOLO: CAMERA [NOME] \cap SENATO [NOME] = \emptyset

RIUNIONI (DATA, SENATO, CATEGORIA)

FK: RIU [SEN] \subseteq SEN [NOME]

COMPOSIZIONE (COM, MESE, ANNO)

FK: COMP [COM] \subseteq COM [NOME]

FK: COMP [COM, M, A] \subseteq PRESIEDE [COM, M, A]

PRESIEDE (COM, MESE, ANNO, GP, PARL)

FK: PR [COM, M, A] \subseteq COMPOSIZIONE [COM, M, A]

FK: PR [PARL] \subseteq PARL [CF]



ACCORPO COMPOSIZIONE CON PRESIEDE

COMPOSIZIONE (NOME, MESE, ANNO, GP, PARL)

FK: COMP [COM] \subseteq COM [NOME]

FK: COMP [PARL] \subseteq PARL [CF]

PROBLEMA 3

1) SELECT C.DOLENTE, COUNT(*) AS CONCORSI
FROM CONCORSO C
GROUP BY C.DOLENTE

UNION

SELECT CODICE, 0
FROM DOLENTE
WHERE CODICE NOT IN (SELECT DISTINCT DOLENTE
FROM CONCORSO)

2) SELECT DISTINCT D.CODICE, MIN(C.PUNTEGGIO)
FROM DOLENTE D JOIN CONCORSO C ON
D.CODICE = C.DOLENTE
WHERE D.CITTA_NASCITA = "ROMA"
GROUP BY D.CODICE

UNION

```

SELECT CODICE, NULL
FROM DOLENTE
WHERE D. DATA NASCITA = "ROMA"
AND D.CODICE NOT IN (SELECT DOLENTE
FROM CONCORSO)

```

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

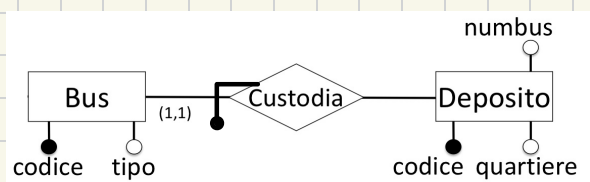
X

PROBLEMA 4

1) PROJ_{CODICE, C}
 ((DOLENTE JOIN CONCORSO)
 CODICE = DOLENTE CODICE != C
 AND
 DATA NASCITA
 = C.N
 REN C ← CODICE
 LN ← DATA NASCITA
 DN ← DATA NASCITA
 D ← DOLENTE
 A ← ANNO
 P ← PUNTEGGIO
 (DOLENTE JOIN CONCORSO)))

2) PROJ_{CODICE}(DOLENTE) - PROJ_{DOLENTE}(SEL PUNTEGGIO ≤ 25
 OR
 PUNTEGGIO == NULL
 (CONCORSO))

PROBLEMA 5



BUS (CODICE, TIPO)

FK: BUS[CODICE] ≤ CUSTODIA[BUS]

DEPOSITO (CODICE, QUARTIERE, NUMBUS)

CUSTODIA (BUS, DEPOSITO)

FK: BUS[BUS] ≤ BUS[COD]

FK: BUS[DEP] ≤ DEP[COD]

ACCORPIAMO BUS E CUSTODIA

BUS (CODICE, TIPO, DEPOSITO)

FK: BUS [DEP] ≤ DEP [COD]

DEPOSITO (CODICE, QUARTIERE, NUMBUS)

CREATE TABLE BUS (

CODICE VARCHAR PRIMARY KEY,

QUARTIERE VARCHAR NOT NULL,

DEPOSITO VARCHAR,

FOREIGN KEY (DEPOSITO) REFERENCES DEPOSITO (CODICE)

)

CREATE TABLE DEPOSITO (

CODICE VARCHAR PRIMARY KEY,

QUARTIERE VARCHAR NOT NULL,

NUMBUS INTEGER NOT NULL,

)