

Matricola: _____

Cognome: _____

Nome: _____

Basi di Dati

Prova di teoria del 19 giugno 2020

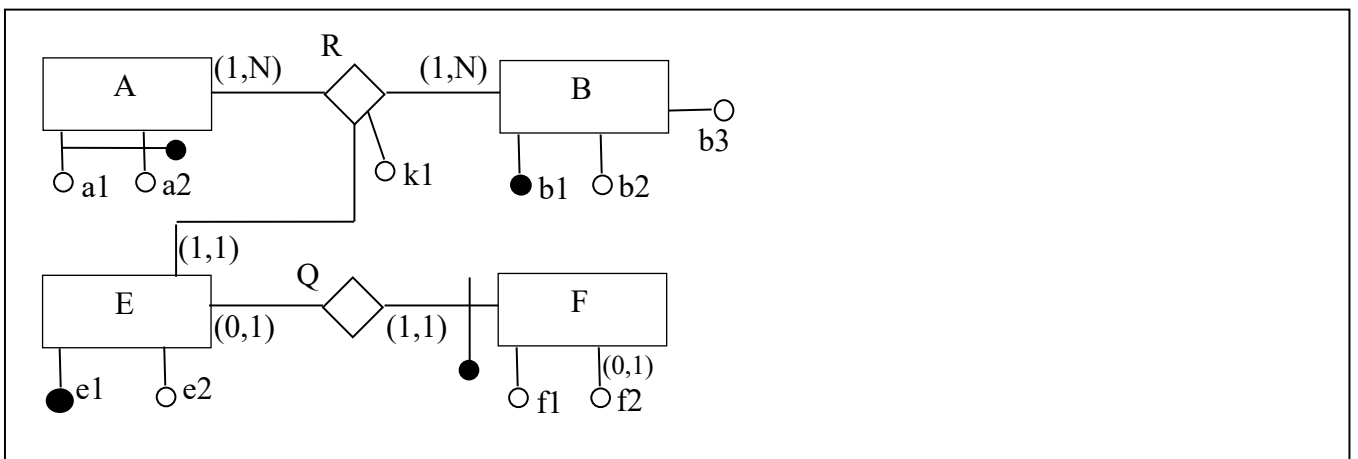
Avvertenze: e' severamente vietato consultare libri e appunti.

Durata 2h10m

DOMANDE PRELIMINARI (è necessario rispondere in modo sufficiente alle seguenti tre domande per poter superare la prova scritta con esito positivo; in caso di mancata o errata risposta a queste domande il resto del compito non verrà corretto)

a) Si illustri il costrutto di relazione del modello Entità-Relazioni

b) Dato il seguente schema concettuale nel modello ER, si produca la sua traduzione nel modello relazionale



c) Date le due seguenti relazioni: $R1(\underline{A}, B, W, Z)$ e $R2(\underline{D}, E, F, G)$ (tutti gli attributi sono di tipo numerico) scrivere:

- un'espressione in algebra relazionale che restituisca le combinazioni distinte di valori contenuti negli attributi W e Z di R1;
- un'espressione ottimizzata dell'algebra relazionale che contenga un theta join, una selezione su R2 e due proiezioni e produca come risultato le tuple t di R2 tali che $0 < t[E] < 10$ e tali che esiste una tupla t' di R1 dove $t[G] > t'[B]$ (non sono ammesse altre selezioni oltre a quella su R2).

Punteggi esercizi: (1) 12 - (2.a, 2.b, 2.c) 3 – (3.a, 3.b) 3 – (4) 2 – (5) 4

È obbligatorio rispondere alle domande 1 e 2 in modo sufficiente.

1. Si vuole progettare un sistema informativo per la gestione delle prenotazioni per assistere alle lezioni in presenza nel prossimo semestre dell'anno accademico 2020/2021 presso l'Area di Scienze e Ingegneria dell'università di Verona. Per ogni insegnamento (o modulo di insegnamento) da erogare nel semestre si memorizza: codice univoco, nome, tipo attività (lezione frontale, laboratorio, esercitazione), corso di studi, anno (ad esempio: 343, Basi di dati, lezione, Informatica, 3° anno). Per ogni insegnamento si memorizzano tutte le lezioni allocate nel calendario indicando oltre all'insegnamento: la data e l'ora della lezione, la durata e l'aula dove si svolge. Per le aule si memorizza: nome, edificio (la coppia nome, edificio identifica un'aula), piano, capienza massima e capienza Covid19.
Gli studenti possono prenotare una lezione in calendario attraverso una applicazione web; per ogni prenotazione l'applicazione registra nel sistema: lo studente, la lezione, la data e l'ora della prenotazione. Il sistema memorizza per ogni studente: idGIA (univoco), nome, cognome, per gli studenti non residenti nel comune di Verona registra inoltre il comune di residenza e la distanza in Km dalla sede dell'erogazione delle lezioni.
All'inizio delle lezioni gli studenti si presentano con il loro badge personale in cui è registrato il proprio idGIA e eseguono una strisciata all'ingresso dell'aula e una strisciata all'uscita dall'aula. Il sistema registra quindi tali strisciate memorizzando: il tipo di strisciata (ingresso o uscita), lo studente, la data e l'ora della strisciata e l'aula. Al termine di ogni mese del semestre si registra il numero totale e il numero medio di prenotazioni ricevute per un insegnamento.
Infine si registrano tutti gli interventi di sanificazione delle aule indicando: data e ora della sanificazione, l'aula interessata, la ditta che ha eseguito il lavoro e i responsabili della sanificazione (si registrano i codici fiscali dei responsabili).
Progettare lo schema concettuale utilizzando il modello entità-relazione e lo schema relazionale della base di dati (indicare esplicitamente per ogni relazione dello schema relazionale: le chiavi primarie, gli attributi che possono contenere valori nulli e i vincoli di integrità referenziale). Non aggiungere attributi non esplicitamente indicati nel testo.
2. Dato il seguente schema relazionale (chiavi primarie sottolineate) le informazioni che riguardano gli esami fatti dai pazienti di un ospedale:
REPARTO(CodReparto, NomeRep, Primario, numeroDipendenti);
ESAME(CodReparto, Paziente, dataEsame, oraEsame, Tipo, Esito, NoteDelMedico, Urgenza:{si,no})
PAZIENTE(TesseraSanitaria, Nome, Cognome, RegioneRes, Nazionalità)
Vincoli di integrità: ESAME.CodReparto → REPARTO, ESAME.Paziente → PAZIENTE
Formulare in algebra relazionale le seguenti interrogazioni:
2.a Trovare i pazienti non di nazionalità "italiana" che ieri hanno fatto un esame di tipo "test glicemico" presso il reparto di "Ematologia" riportando il cognome, il nome del paziente, l'esito dell'esame e le note del medico.
2.b Trovare i reparti che nel mese di Gennaio 2020 non hanno fatto esami di tipo "emoglobina" a pazienti residenti in "Veneto", riportando il nome del reparto e il primario.
2.c Trovare il nome e il cognome dei pazienti che hanno fatto almeno due esami di tipo "test glicemico" in due giorni diversi del mese di Dicembre 2019.
3. Dato lo schema dell'esercizio 2:
 - a. Scrivere in SQL la seguente interrogazione:
Analizzare gli esami urgenti fatti nel mese di Giugno 2019 e trovare per ogni tipo esame con almeno 5 esami urgenti fatti: il numero totale di pazienti che hanno fatto esami urgenti di quel tipo e il numero di regioni di residenza distinte degli stessi pazienti riportando nel risultato: il tipo di esame e i conteggi richiesti.
 - b. Data la seguente interrogazione:

```
SELECT Nome, Cognome, dataEsame, oraEsame
FROM PAZIENTE JOIN ESAME ON TesseraSanitaria = Paziente WHERE RegioneRes <> 'Veneto'
```


Indicare una stima del costo dell'interrogazione in termini di numero di accessi a memoria secondaria sapendo che: (i) la selezione dei pazienti viene eseguita attraverso una scansione della tabella PAZIENTE e il risultato viene mantenuto nel buffer, (ii) l'ordine di esecuzione del join è PAZIENTE ⋈ ESAME e viene applicata la tecnica "Nested Loop Join" e con indice B+tree di profondità 3 sull'attributo Paziente della tabella ESAME; (iii) NP(PAZIENTE)=15, NR(PAZIENTE) = 1900, NP(ESAME) = 155, NR(ESAME) = 38000, VAL(Regione, PAZIENTE) = 19 e VAL(TesseraSanitaria, ESAME) = 1900.
4. Lo studente illustri il concetto di view-equivalenza tra schedule.

5. Specificare in XML Schema la sintassi del seguente documento XML. Vincoli: l'attributo *data* è obbligatorio, l'attributo *tessera* identifica il paziente, l'elemento *tipo* può contenere solo i valori: emoglobina, colesterolo, glicemia.

```
<esami xmlns="http://www.esami.org
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.esami.org esami.xsd">
  <giorno data="12/12/2019">
    <esame><paziente tessera="XXX01">
      <nome>Paolo</nome><cognome>Rossi</cognome>
    </paziente>
    <tipo>emoglobina</tipo><esito valore="10">note del medico</esito>
  </esame>
</giorno>
<giorno data="13/07/2018">
  <esame><paziente tessera="XXX02"><cognome>Bianchi</cognome></paziente>
    <tipo>glicemia</tipo><esito valore="90">note del medico</esito>
  </esame>
  <esame><paziente tessera="XXX03"><cognome>Verdi</cognome></paziente>
    <tipo>colesterolo</tipo><esito valore="110">note del medico</esito>
  </esame>
</giorno>
</esami>
```

XMLSchema Esami.xsd da completare

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
(1)...
<xsd:element name="paziente">
  <xsd:complexType> <xsd:sequence>
    <xsd:element (2)... />
    <xsd:element (3)... />
  </xsd:sequence>
  <xsd:attribute (4)... />
</xsd:complexType>
</xsd:element>
<xsd:simpleType name="ST_tipoEsame">
  <xsd:restriction base="...(5)" ">
    ...(6)
    ...
    ...
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name="esito">
<xsd:complexType><xsd:simpleContent>
  <xsd:extension ...(7)
    <xsd:attribute ...(8) />
  </xsd:extension>
</xsd:simpleContent></xsd:complexType>
</xsd:element>
<xsd:complexType name="CT_Esame">
  <xsd:sequence>
    <xsd:element ...(9) />
    <xsd:element ... />
    <xsd:element ... />
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="giorno">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ...(10) />
    </xsd:sequence>
    <xsd:attribute ...(11) />
  </xsd:complexType>
</xsd:element>
<xsd:element name="esami">
... ...(12)
...
...
</xsd:element>
</xsd:schema>
```