Tecnologia delle basi di dati (ex Basi di dati, primo modulo)

15 giugno 2005 — Compito A

Tempo a disposizione: due ore e quindici minuti

Domanda 1 (25%) Siano r_1 ed r_2 due relazioni contenenti rispettivamente N_1 e N_2 ennuple, con fattore di blocco rispettivamente F_1 e F_2. Si supponga che il sistema abbia a disposizione un buffer di dimensione pari a $M = 101$ blocchi. Calcolare il numero di accessi a memoria secondaria necessario per eseguire un join $r_1 \bowtie_{A_1=A_2} r_2$ (con A_1 attributo di r_1 e A_2 attributo di r_2), nei seguenti casi, da considerare separatamente l'uno dall'altro e assumendo che il DBMS sia in grado di eseguire il join solo con il metodo *nested-loop* (eventualmente utilizzando l'accesso diretto tramite un indice invece della scansione interna) e che utilizzi solo strutture primarie disordinate. Si supponga infine che il blocco abbia dimensioni $B =$ 1 Kbyte, che i puntatori occupino $p = 4$ byte e i valori dei due attributi in questione occupino $k = 20$ byte ciascuno.

1. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_2 è la chiave di r_2 mentre i valori di A_1 in r_1 si ripetono mediamente in $e = 6$ ennuple; non vi sono indici
2. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_1 è la chiave di r_1 mentre i valori di A_2 in r_2 si ripetono mediamente in $e = 6$ ennuple; sono definiti un indice su $r_1(A_1)$ e uno su $r_2(A_2)$
3. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; gli attributi coinvolti non sono chiave e hanno, ciascuno, valori che si ripetono mediamente in $e = 6$ ennuple; è definito un indice su $r_1(A_1)$
4. $N_1 = 5.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 20$; A_1 è la chiave di r_1 e A_2 è la chiave di r_2; sono definiti un indice su $r_1(A_1)$ e uno su $r_2(A_2)$

Domanda 2 (20%) Si consideri la base di dati seguente, relativa ai voli effettuati dagli iscritti ad un programma fedeltà ("frequent flyer") di una compagnia aerea (vengono indicati, sia pure in modo informale, i vincoli di riferimento):

- Clienti(CodiceCliente,...,Status); nota: lo "Status" indica la categoria attuale dell'iscritto al programma (ad esempio, "normale," "silver," "gold"), che è variabile nel tempo, con aggiornamenti fatti con periodicità fissa, ad esempio mensile e qui rappresentato semplicemente per mezzo di una stringa; omettiamo gli altri dati, assumendo che non interessino.
- Biglietti(NumeroBiglietto,Cliente,DataAcquisto,NumeroVolo,DataVolo), con vincoli di riferimento verso Clienti e verso VoloSpecifico
- VoloSpecifico(NumeroVolo,DataVolo,Ritardo) con vincolo di riferimento verso VoloAstratto
- VoloAstratto(NumeroVolo,Origine,Destinazione,Aeromobile) con vincoli di riferimento verso Aeroporto (due vincoli diversi) e verso Aeromobile (si suppone che l'aeromobile di un volo astratto sia usato per tutti i voli specifici e non cambi).
- Aeromobile(CodiceAeromobile,Descrizione)
- Aeroporto(CodiceAeroporto,Città)

Con riferimento a tale base di dati, progettare uno schema dimensionale che permetta di rispondere facilmente ad interrogazioni quali ad esempio (la lista non ha pretesa di essere esaustiva):

- trovare le destinazioni (sia in termini di città sia di aree geografiche, quali nazione e continente) più importanti per gli iscritti al programma, per ciascuno di essi e per ciascuna categoria ("Status")
- gli aeromobili utilizzati dagli iscritti al programma
- il ritardo medio subito da ciascun iscritto e dagli iscritti di ciascuna categoria
- quanto in anticipo vengono comprati i biglietti, anche con riferimento alle varie stagioni dell'anno e in particolare ai periodi di vacanza.

Allo scopo:

1. specificare un possibile dettaglio dello schema dimensionale utilizzato (uno solo);
2. individuare quali dati siano necessari nel data mart e non presenti nelle basi di dati operative;
3. supponendo che l'alimentazione del data mart sia quotidiana, specificare le operazioni (ad esempio in termini di istruzioni SQL o di espressioni dell'algebra relazionale) da svolgere allo scopo.

Domanda 3 (10%) Con riferimento alla domanda precedente e supponendo che la relazione VoloSpecifico abbia anche un attributo NumeroPasseggeri (con valore pari al numero di passeggeri complessivamente presenti sul volo stesso, inclusi quindi anche quelli non iscritti al programma fedeltà) mostrare anche un secondo schema dimensionale che permetta di correlare il comportamento degli iscritti con quello della clientela in senso lato. Si noti che per i non iscritti al programma non sono note informazioni di dettaglio né sulle persone né sui biglietti.

Domanda 4 (20%) Una catena di supermercati ha una base di dati dei propri clienti che dispongono di una "tessera fedeltà," con varie informazioni su ciascun cliente, fra cui (a) il totale dei punti acquisiti attraverso l'uso della tessera e (b) il negozio della catena cui fa riferimento (ad esempio, quello presso cui ha inizialmente richiesto la tessera). Si vuole eseguire su di essa l'interrogazione che calcola, per ciascun negozio, il numero dei clienti, la somma dei punti fedeltà dei clienti e la relativa media per cliente. Indicare (con un breve commento, non più di tre righe) quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sbagliati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. L'operazione è eseguita in un momento in cui non ci sono aggiornamenti di alcun genere, con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
2. L'operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
3. L'operazione è eseguita mentre vengono inseriti molti nuovi clienti, con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
4. L'operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
5. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.

Domanda 5 (25%) Una tecnica per il controllo di concorrenza diversa da quelle viste nel corso va sotto il nome di "concorrenza basata sulla validazione" ed è basata sulle seguenti ipotesi e princìpi.

- Ogni transazione è divisa in tre fasi: (i) prima esegue tutte le proprie letture e svolge le proprie elaborazioni "privatamente" (cioè in memoria, senza scrivere sulla base di dati); (ii) poi chiede allo scheduler il permesso di andare in commit (fase di "validazione") e (iii) se autorizzata, scrive.
- Lo scheduler ricorda per ogni transazione (1) gli insiemi di dati che essa ha letto $\textsc{rSet}(T)$ e ha scritto (o vuole scrivere) $\textsc{wSet}(T)$, (2) in quale "stato" si trova: INIZ (iniziata e non ancora validata), VAL (validata, ma con scritture da completare), CPL (completata, con l'esecuzione di tutte le scritture) e (3) i corrispondenti istanti di inizio $\textsc{iniz}(T)$, validazione $\textsc{val}(T)$ e completamento $\textsc{cpl}(T)$.
- Ad una transazione T la validazione viene concessa in tutti i casi esclusi i seguenti, in cui viene negata:
  1. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata o completata, che non era completata quando T è iniziata (e quindi risulterebbe $\textsc{cpl}(T') > \textsc{iniz}(T)$)
     - un dato $x \in \textsc{rSet}(T) \cap \textsc{wSet}(T')$
  2. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata ma non completata (e quindi risulterebbe $\textsc{cpl}(T') > t = \textsc{val}(T)$)
     - un dato $x \in \textsc{wSet}(T) \cap \textsc{wSet}(T')$

1. dimostrare (formalmente o almeno intuitivamente) che la classe di schedule prodotta è propriamente contenuta in CSR.
2. individuare quando possono essere eliminate le informazioni su una transazione, al fine di evitare la crescita eccessiva dei dati da gestire.

Tecnologia delle basi di dati (ex Basi di dati, primo modulo)

15 giugno 2005 — Compito B

Tempo a disposizione: due ore e quindici minuti

Domanda 1 (25%) Siano r_1 ed r_2 due relazioni contenenti rispettivamente N_1 e N_2 ennuple, con fattore di blocco rispettivamente F_1 e F_2. Si supponga che il sistema abbia a disposizione un buffer di dimensione pari a $M = 101$ blocchi. Calcolare il numero di accessi a memoria secondaria necessario per eseguire un join $r_1 \bowtie_{A_1=A_2} r_2$ (con A_1 attributo di r_1 e A_2 attributo di r_2), nei seguenti casi, da considerare separatamente l'uno dall'altro e assumendo che il DBMS sia in grado di eseguire il join solo con il metodo *nested-loop* (eventualmente utilizzando l'accesso diretto tramite un indice invece della scansione interna) e che utilizzi solo strutture primarie disordinate. Si supponga infine che il blocco abbia dimensioni $B =$ 1 Kbyte, che i puntatori occupino $p = 4$ byte e i valori dei due attributi in questione occupino $k = 20$ byte ciascuno.

1. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_2 è la chiave di r_2 mentre i valori di A_1 in r_1 si ripetono mediamente in $e = 6$ ennuple; è definito un indice su $r_1(A_1)$
2. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; gli attributi coinvolti non sono chiave e hanno, ciascuno, valori che si ripetono mediamente in $e = 6$ ennuple; sono definiti un indice su $r_1(A_1)$ e uno su $r_2(A_2)$
3. $N_1 = 5.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 20$; A_1 è la chiave di r_1 e A_2 è la chiave di r_2; è definito un indice su $r_1(A_1)$
4. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; gli attributi coinvolti non sono chiave e hanno, ciascuno, valori che si ripetono mediamente in $e = 6$ ennuple; non vi sono indici

Domanda 2 (20%) Si consideri la base di dati seguente, relativa ai voli effettuati dagli iscritti ad un programma fedeltà ("frequent flyer") di una compagnia aerea (vengono indicati, sia pure in modo informale, i vincoli di riferimento):

- Soci(NumTessera,...,Status); nota: lo "Status" indica la categoria attuale dell'iscritto al programma (ad esempio, "normale," "silver," "gold"), che è variabile nel tempo, con aggiornamenti fatti con periodicità fissa, ad esempio mensile e qui rappresentato semplicemente per mezzo di una stringa; omettiamo gli altri dati, assumendo che non interessino.
- Biglietti(NumeroBiglietto,Socio,DataAcquisto,NumeroVolo,DataVolo), con vincoli di riferimento verso Soci e verso VoloSpecifico
- VoloSpecifico(NumeroVolo,DataVolo,Ritardo) con vincolo di riferimento verso VoloAstratto
- VoloAstratto(NumeroVolo,Origine,Destinazione,Aeromobile) con vincoli di riferimento verso Aeroporto (due vincoli diversi) e verso Aeromobile (si suppone che l'aeromobile di un volo astratto sia usato per tutti i voli specifici e non cambi).
- Aeromobile(CodiceAeromobile,Descrizione)
- Aeroporto(CodiceAeroporto,Città)

Con riferimento a tale base di dati, progettare uno schema dimensionale che permetta di rispondere facilmente ad interrogazioni quali ad esempio (la lista non ha pretesa di essere esaustiva):

- trovare le destinazioni (sia in termini di città sia di aree geografiche, quali nazione e continente) più importanti per gli iscritti al programma, per ciascuno di essi e per ciascuna categoria ("Status")
- gli aeromobili utilizzati dagli iscritti al programma
- il ritardo medio subito da ciascun iscritto e dagli iscritti di ciascuna categoria
- quanto in anticipo vengono comprati i biglietti, anche con riferimento alle varie stagioni dell'anno e in particolare ai periodi di vacanza.

Allo scopo:

1. specificare un possibile dettaglio dello schema dimensionale utilizzato (uno solo);
2. individuare quali dati siano necessari nel data mart e non presenti nelle basi di dati operative;
3. supponendo che l'alimentazione del data mart sia quotidiana, specificare le operazioni (ad esempio in termini di istruzioni SQL o di espressioni dell'algebra relazionale) da svolgere allo scopo.

Domanda 3 (10%) Con riferimento alla domanda precedente e supponendo che la relazione VoloSpecifico abbia anche un attributo NumeroPasseggeri (con valore pari al numero di passeggeri complessivamente presenti sul volo stesso, inclusi quindi anche quelli non iscritti al programma fedeltà) mostrare anche un secondo schema dimensionale che permetta di correlare il comportamento degli iscritti con quello della clientela in senso lato. Si noti che per i non iscritti al programma non sono note informazioni di dettaglio né sulle persone né sui biglietti.

Domanda 4 (20%) Una catena di supermercati ha una base di dati dei propri clienti che dispongono di una "tessera fedeltà," con varie informazioni su ciascun cliente, fra cui (a) il totale dei punti acquisiti attraverso l'uso della tessera e (b) il negozio della catena cui fa riferimento (ad esempio, quello presso cui ha inizialmente richiesto la tessera). Si vuole eseguire su di essa l'interrogazione che calcola, per ciascun negozio, il numero dei clienti, la somma dei punti fedeltà dei clienti e la relativa media per cliente. Indicare (con un breve commento, non più di tre righe) quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sbagliati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. L'operazione è eseguita mentre vengono inseriti molti nuovi clienti, con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
2. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di alcuni clienti (in ciascuna agenzia una piccola percentuale), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
3. L'operazione è eseguita in un momento in cui non ci sono aggiornamenti di alcun genere, con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
4. L'operazione è eseguita mentre vengono inseriti molti nuovi clienti, con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
5. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di alcuni clienti (in ciascuna agenzia una piccola percentuale), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.

Domanda 5 (25%) Una tecnica per il controllo di concorrenza diversa da quelle viste nel corso va sotto il nome di "concorrenza basata sulla validazione" ed è basata sulle seguenti ipotesi e princìpi.

- Ogni transazione è divisa in tre fasi: (i) prima esegue tutte le proprie letture e svolge le proprie elaborazioni "privatamente" (cioè in memoria, senza scrivere sulla base di dati); (ii) poi chiede allo scheduler il permesso di andare in commit (fase di "validazione") e (iii) se autorizzata, scrive.
- Lo scheduler ricorda per ogni transazione (1) gli insiemi di dati che essa ha letto $\textsc{rSet}(T)$ e ha scritto (o vuole scrivere) $\textsc{wSet}(T)$, (2) in quale "stato" si trova: STRT (iniziata e non ancora validata), VAL (validata, ma con scritture da completare), FIN (completata, con l'esecuzione di tutte le scritture) e (3) i corrispondenti istanti di inizio $\textsc{strt}(T)$, validazione $\textsc{val}(T)$ e completamento $\textsc{fin}(T)$.
- Ad una transazione T la validazione viene concessa in tutti i casi esclusi i seguenti, in cui viene negata:
  1. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata o completata, che non era completata quando T è iniziata (e quindi risulterebbe $\textsc{fin}(T') > \textsc{strt}(T)$)
     - un dato $x \in \textsc{rSet}(T) \cap \textsc{wSet}(T')$
  2. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata ma non completata (e quindi risulterebbe $\textsc{fin}(T') > t = \textsc{val}(T)$)
     - un dato $x \in \textsc{wSet}(T) \cap \textsc{wSet}(T')$

1. dimostrare (formalmente o almeno intuitivamente) che la classe di schedule prodotta è propriamente contenuta in CSR.
2. individuare quando possono essere eliminate le informazioni su una transazione, al fine di evitare la crescita eccessiva dei dati da gestire.

Tecnologia delle basi di dati (ex Basi di dati, primo modulo)

15 giugno 2005 — Compito C

Tempo a disposizione: due ore e quindici minuti

Domanda 1 (25%) Siano r_1 ed r_2 due relazioni contenenti rispettivamente N_1 e N_2 ennuple, con fattore di blocco rispettivamente F_1 e F_2. Si supponga che il sistema abbia a disposizione un buffer di dimensione pari a $M = 101$ blocchi. Calcolare il numero di accessi a memoria secondaria necessario per eseguire un join $r_1 \bowtie_{A_1=A_2} r_2$ (con A_1 attributo di r_1 e A_2 attributo di r_2), nei seguenti casi, da considerare separatamente l'uno dall'altro e assumendo che il DBMS sia in grado di eseguire il join solo con il metodo *nested-loop* (eventualmente utilizzando l'accesso diretto tramite un indice invece della scansione interna) e che utilizzi solo strutture primarie disordinate. Si supponga infine che il blocco abbia dimensioni $B =$ 1 Kbyte, che i puntatori occupino $p = 4$ byte e i valori dei due attributi in questione occupino $k = 20$ byte ciascuno.

1. $N_1 = 5.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 20$; gli attributi coinvolti non sono chiave e hanno, ciascuno, valori che si ripetono mediamente in $e = 6$ ennuple; sono definiti un indice su $r_1(A_1)$ e uno su $r_2(A_2)$
2. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_1 è la chiave di r_1 mentre i valori di A_2 in r_2 si ripetono mediamente in $e = 6$ ennuple; non vi sono indici
3. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_1 è la chiave di r_1 e A_2 è la chiave di r_2; sono definiti un indice su $r_1(A_1)$ e uno su $r_2(A_2)$
4. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_1 è la chiave di r_1 e A_2 è la chiave di r_2; è definito un indice su $r_1(A_1)$

Domanda 2 (20%) Si consideri la base di dati seguente, relativa ai voli effettuati dagli iscritti ad un programma fedeltà ("frequent flyer") di una compagnia aerea (vengono indicati, sia pure in modo informale, i vincoli di riferimento):

- Soci(NumTessera,...,Status); nota: lo "Status" indica la categoria attuale dell'iscritto al programma (ad esempio, "normale," "silver," "gold"), che è variabile nel tempo, con aggiornamenti fatti con periodicità fissa, ad esempio mensile e qui rappresentato semplicemente per mezzo di una stringa; omettiamo gli altri dati, assumendo che non interessino.
- Biglietti(NumeroBiglietto,Socio,DataAcquisto,NumeroVolo,DataVolo), con vincoli di riferimento verso Soci e verso VoloSpecifico
- VoloSpecifico(NumeroVolo,DataVolo,Ritardo) con vincolo di riferimento verso VoloAstratto
- VoloAstratto(NumeroVolo,Origine,Destinazione,Aeromobile) con vincoli di riferimento verso Aeroporto (due vincoli diversi) e verso Aeromobile (si suppone che l'aeromobile di un volo astratto sia usato per tutti i voli specifici e non cambi).
- Aeromobile(CodiceAeromobile,Descrizione)
- Aeroporto(CodiceAeroporto,Città)

Con riferimento a tale base di dati, progettare uno schema dimensionale che permetta di rispondere facilmente ad interrogazioni quali ad esempio (la lista non ha pretesa di essere esaustiva):

- trovare le destinazioni (sia in termini di città sia di aree geografiche, quali nazione e continente) più importanti per gli iscritti al programma, per ciascuno di essi e per ciascuna categoria ("Status")
- gli aeromobili utilizzati dagli iscritti al programma
- il ritardo medio subito da ciascun iscritto e dagli iscritti di ciascuna categoria
- quanto in anticipo vengono comprati i biglietti, anche con riferimento alle varie stagioni dell'anno e in particolare ai periodi di vacanza.

Allo scopo:

1. specificare un possibile dettaglio dello schema dimensionale utilizzato (uno solo);
2. individuare quali dati siano necessari nel data mart e non presenti nelle basi di dati operative;
3. supponendo che l'alimentazione del data mart sia quotidiana, specificare le operazioni (ad esempio in termini di istruzioni SQL o di espressioni dell'algebra relazionale) da svolgere allo scopo.

Domanda 3 (10%) Con riferimento alla domanda precedente e supponendo che la relazione VoloSpecifico abbia anche un attributo NumeroPasseggeri (con valore pari al numero di passeggeri complessivamente presenti sul volo stesso, inclusi quindi anche quelli non iscritti al programma fedeltà) mostrare anche un secondo schema dimensionale che permetta di correlare il comportamento degli iscritti con quello della clientela in senso lato. Si noti che per i non iscritti al programma non sono note informazioni di dettaglio né sulle persone né sui biglietti.

Domanda 4 (20%) Una catena di supermercati ha una base di dati dei propri clienti che dispongono di una "tessera fedeltà," con varie informazioni su ciascun cliente, fra cui (a) il totale dei punti acquisiti attraverso l'uso della tessera e (b) il negozio della catena cui fa riferimento (ad esempio, quello presso cui ha inizialmente richiesto la tessera). Si vuole eseguire su di essa l'interrogazione che calcola, per ciascun negozio, il numero dei clienti, la somma dei punti fedeltà dei clienti e la relativa media per cliente. Indicare (con un breve commento, non più di tre righe) quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sbagliati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. L'operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
2. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
3. L'operazione è eseguita in un momento in cui non ci sono aggiornamenti di alcun genere, con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
4. L'operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascun negozio pochi rispetto a quelli già presenti), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
5. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.

Domanda 5 (25%) Una tecnica per il controllo di concorrenza diversa da quelle viste nel corso va sotto il nome di "concorrenza basata sulla validazione" ed è basata sulle seguenti ipotesi e princìpi.

- Ogni transazione è divisa in tre fasi: (i) prima esegue tutte le proprie letture e svolge le proprie elaborazioni "privatamente" (cioè in memoria, senza scrivere sulla base di dati); (ii) poi chiede allo scheduler il permesso di andare in commit (fase di "validazione") e (iii) se autorizzata, scrive.
- Lo scheduler ricorda per ogni transazione (1) gli insiemi di dati che essa ha letto $\textsc{rSet}(T)$ e ha scritto (o vuole scrivere) $\textsc{wSet}(T)$, (2) in quale "stato" si trova: STRT (iniziata e non ancora validata), VAL (validata, ma con scritture da completare), FIN (completata, con l'esecuzione di tutte le scritture) e (3) i corrispondenti istanti di inizio $\textsc{strt}(T)$, validazione $\textsc{val}(T)$ e completamento $\textsc{fin}(T)$.
- Ad una transazione T la validazione viene concessa in tutti i casi esclusi i seguenti, in cui viene negata:
  1. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata o completata, che non era completata quando T è iniziata (e quindi risulterebbe $\textsc{fin}(T') > \textsc{strt}(T)$)
     - un dato $x \in \textsc{rSet}(T) \cap \textsc{wSet}(T')$
  2. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata ma non completata (e quindi risulterebbe $\textsc{fin}(T') > t = \textsc{val}(T)$)
     - un dato $x \in \textsc{wSet}(T) \cap \textsc{wSet}(T')$

1. dimostrare (formalmente o almeno intuitivamente) che la classe di schedule prodotta è propriamente contenuta in CSR.
2. individuare quando possono essere eliminate le informazioni su una transazione, al fine di evitare la crescita eccessiva dei dati da gestire.

Tecnologia delle basi di dati (ex Basi di dati, primo modulo)

15 giugno 2005 — Compito D

Tempo a disposizione: due ore e quindici minuti

Domanda 1 (25%) Siano r_1 ed r_2 due relazioni contenenti rispettivamente N_1 e N_2 ennuple, con fattore di blocco rispettivamente F_1 e F_2. Si supponga che il sistema abbia a disposizione un buffer di dimensione pari a $M = 101$ blocchi. Calcolare il numero di accessi a memoria secondaria necessario per eseguire un join $r_1 \bowtie_{A_1=A_2} r_2$ (con A_1 attributo di r_1 e A_2 attributo di r_2), nei seguenti casi, da considerare separatamente l'uno dall'altro e assumendo che il DBMS sia in grado di eseguire il join solo con il metodo *nested-loop* (eventualmente utilizzando l'accesso diretto tramite un indice invece della scansione interna) e che utilizzi solo strutture primarie disordinate. Si supponga infine che il blocco abbia dimensioni $B =$ 1 Kbyte, che i puntatori occupino $p = 4$ byte e i valori dei due attributi in questione occupino $k = 20$ byte ciascuno.

1. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_2 è la chiave di r_2 mentre i valori di A_1 in r_1 si ripetono mediamente in $e = 6$ ennuple; sono definiti un indice su $r_1(A_1)$ e uno su $r_2(A_2)$
2. $N_1 = 5.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 20$; gli attributi coinvolti non sono chiave e hanno, ciascuno, valori che si ripetono mediamente in $e = 6$ ennuple; è definito un indice su $r_1(A_1)$
3. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_1 è la chiave di r_1 mentre i valori di A_2 in r_2 si ripetono mediamente in $e = 6$ ennuple; è definito un indice su $r_1(A_1)$
4. $N_1 = 100.000$ e $N_2 = 1.000.000$, con $F_1 = F_2 = 10$; A_1 è la chiave di r_1 e A_2 è la chiave di r_2; non vi sono indici

Domanda 2 (20%) Si consideri la base di dati seguente, relativa ai voli effettuati dagli iscritti ad un programma fedeltà ("frequent flyer") di una compagnia aerea (vengono indicati, sia pure in modo informale, i vincoli di riferimento):

- Soci(NumTessera,...,Status); nota: lo "Status" indica la categoria attuale dell'iscritto al programma (ad esempio, "normale," "silver," "gold"), che è variabile nel tempo, con aggiornamenti fatti con periodicità fissa, ad esempio mensile e qui rappresentato semplicemente per mezzo di una stringa; omettiamo gli altri dati, assumendo che non interessino.
- Biglietti(NumeroBiglietto,Socio,DataAcquisto,NumeroVolo,DataVolo), con vincoli di riferimento verso Soci e verso VoloSpecifico
- VoloSpecifico(NumeroVolo,DataVolo,Ritardo) con vincolo di riferimento verso VoloAstratto
- VoloAstratto(NumeroVolo,Origine,Destinazione,Aeromobile) con vincoli di riferimento verso Aeroporto (due vincoli diversi) e verso Aeromobile (si suppone che l'aeromobile di un volo astratto sia usato per tutti i voli specifici e non cambi).
- Aeromobile(CodiceAeromobile,Descrizione)
- Aeroporto(CodiceAeroporto,Città)

Con riferimento a tale base di dati, progettare uno schema dimensionale che permetta di rispondere facilmente ad interrogazioni quali ad esempio (la lista non ha pretesa di essere esaustiva):

- trovare le destinazioni (sia in termini di città sia di aree geografiche, quali nazione e continente) più importanti per gli iscritti al programma, per ciascuno di essi e per ciascuna categoria ("Status")
- gli aeromobili utilizzati dagli iscritti al programma
- il ritardo medio subito da ciascun iscritto e dagli iscritti di ciascuna categoria
- quanto in anticipo vengono comprati i biglietti, anche con riferimento alle varie stagioni dell'anno e in particolare ai periodi di vacanza.

Allo scopo:

1. specificare un possibile dettaglio dello schema dimensionale utilizzato (uno solo);
2. individuare quali dati siano necessari nel data mart e non presenti nelle basi di dati operative;
3. supponendo che l'alimentazione del data mart sia quotidiana, specificare le operazioni (ad esempio in termini di istruzioni SQL o di espressioni dell'algebra relazionale) da svolgere allo scopo.

Domanda 3 (10%) Con riferimento alla domanda precedente e supponendo che la relazione VoloSpecifico abbia anche un attributo NumeroPasseggeri (con valore pari al numero di passeggeri complessivamente presenti sul volo stesso, inclusi quindi anche quelli non iscritti al programma fedeltà) mostrare anche un secondo schema dimensionale che permetta di correlare il comportamento degli iscritti con quello della clientela in senso lato. Si noti che per i non iscritti al programma non sono note informazioni di dettaglio né sulle persone né sui biglietti.

Domanda 4 (20%) Una catena di supermercati ha una base di dati dei propri clienti che dispongono di una "tessera fedeltà," con varie informazioni su ciascun cliente, fra cui (a) il totale dei punti acquisiti attraverso l'uso della tessera e (b) il negozio della catena cui fa riferimento (ad esempio, quello presso cui ha inizialmente richiesto la tessera). Si vuole eseguire su di essa l'interrogazione che calcola, per ciascun negozio, il numero dei clienti, la somma dei punti fedeltà dei clienti e la relativa media per cliente. Indicare (con un breve commento, non più di tre righe) quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sbagliati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. L'operazione è eseguita in un momento in cui non ci sono aggiornamenti di alcun genere, con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
2. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di alcuni clienti (in ciascuna agenzia una piccola percentuale), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
3. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
4. L'operazione è eseguita mentre vengono inseriti molti nuovi clienti, con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
5. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di alcuni clienti (in ciascuna agenzia una piccola percentuale), con la finalità di individuare i primi tre negozi da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.

Domanda 5 (25%) Una tecnica per il controllo di concorrenza diversa da quelle viste nel corso va sotto il nome di "concorrenza basata sulla validazione" ed è basata sulle seguenti ipotesi e princìpi.

- Ogni transazione è divisa in tre fasi: (i) prima esegue tutte le proprie letture e svolge le proprie elaborazioni "privatamente" (cioè in memoria, senza scrivere sulla base di dati); (ii) poi chiede allo scheduler il permesso di andare in commit (fase di "validazione") e (iii) se autorizzata, scrive.
- Lo scheduler ricorda per ogni transazione (1) gli insiemi di dati che essa ha letto $\textsc{rSet}(T)$ e ha scritto (o vuole scrivere) $\textsc{wSet}(T)$, (2) in quale "stato" si trova: STRT (iniziata e non ancora validata), VAL (validata, ma con scritture da completare), FIN (completata, con l'esecuzione di tutte le scritture) e (3) i corrispondenti istanti di inizio $\textsc{strt}(T)$, validazione $\textsc{val}(T)$ e completamento $\textsc{fin}(T)$.
- Ad una transazione T la validazione viene concessa in tutti i casi esclusi i seguenti, in cui viene negata:
  1. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata o completata, che non era completata quando T è iniziata (e quindi risulterebbe $\textsc{fin}(T') > \textsc{strt}(T)$)
     - un dato $x \in \textsc{rSet}(T) \cap \textsc{wSet}(T')$
  2. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata ma non completata (e quindi risulterebbe $\textsc{fin}(T') > t = \textsc{val}(T)$)
     - un dato $x \in \textsc{wSet}(T) \cap \textsc{wSet}(T')$

1. dimostrare (formalmente o almeno intuitivamente) che la classe di schedule prodotta è propriamente contenuta in CSR.
2. individuare quando possono essere eliminate le informazioni su una transazione, al fine di evitare la crescita eccessiva dei dati da gestire.