

Sistemi Informativi T
7 luglio 2025

Tempo a disposizione: 3:00 ore

Consegnare i file specificati per gli esercizi 2, 3 e 4. Per l'es. 1 vedere al relativo punto
Si prega di attenersi scrupolosamente alle istruzioni relative ai nomi dei file. **Il mancato rispetto delle istruzioni potrà comportare penalizzazioni di punteggio**
Per superare la prova di è necessario totalizzare **almeno 3 punti negli esercizi 1 e 2**

1) Algebra relazionale (3 punti totali):

Consegnare le risposte su un foglio di carta, intestato con matricola, nome e cognome

Date le seguenti relazioni, definite nello schema B16884 con dati fittizi di esempio:

```
PRODOTTI (ProdID, Categoria, Prezzo);
RECLAMI (RID, ProdID, Data, Cliente, Motivo),
    ProdID REFERENCES PRODOTTI;
ESITI (RID, DataEsito, Esito, Rimborso*),
    RID REFERENCES RECLAMI;
-- DataEsito = la data in cui si è deciso come trattare il reclamo
-- Esito = descrive se e come il reclamo è stato accolto
-- Se Esito = 'RIMBORSO', allora l'attributo Rimborso riporta l'importo
--   rimborsato al cliente (minore o uguale del prezzo del prodotto),
--   altrimenti Rimborso è NULL
-- Prezzo e Rimborso sono di tipo DEC(6,2)
```

si esprimano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** I dati degli esiti di reclami del 2024 che sono stati rimborsati per almeno il 70% del prezzo del prodotto in meno di 20 giorni
- 1.2) [2 p.]** I dati dei prodotti con almeno un reclamo senza esito definito
-

2) SQL (5 punti totali)

*Consegnare il file **ESE2.sql***

Con riferimento al DB dell'esercizio 1, si esprimano in SQL le seguenti interrogazioni:

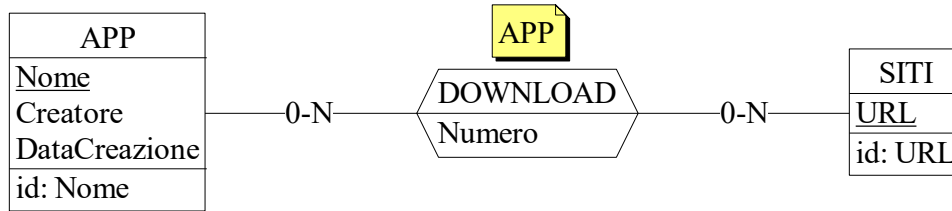
- 2.1) [2 p.]** Per ogni categoria, la media del rapporto Rimborso/Prezzo per i soli prodotti con più di un reclamo
- 2.2) [3 p.]** La categoria per cui i reclami con esito definito sono stati in media più veloci

NB: Per l'uso delle funzioni SQL relative a date, orari e altro si consulti il file FunzioniSQL (disponibile sulla pagina del Lab e su Virtuale)

3) Modifica di schema E/R e del DB (6 punti totali)

Consegnare i file **ESE3.lun** e **ESE3.sql**

Dato il file ESE3.lun fornito, in cui è presente lo schema ESE3-input in figura:



Specifiche aggiuntive:

Si consideri che ogni App può avere diverse VERSIONI, ognuna con la sua data di creazione e numero di download (che quindi non riguardano più l'App in quanto tale)

Traduzione: si traduca tutto ad eccezione di SITI

Operazioni: Si crei una nuova versione di un'App in data odierna, ponendo automaticamente a 0 il numero di download in tutti i siti in cui sono già presenti precedenti versioni della stessa App

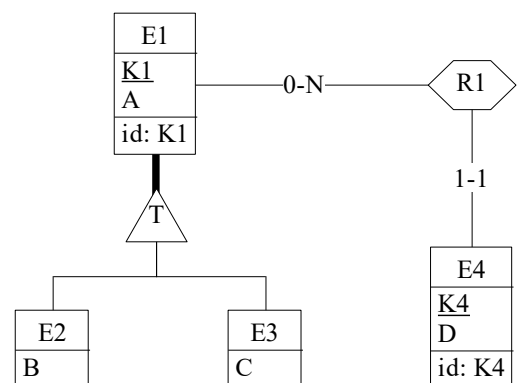
- 3.1) [2 p.] Si copi lo schema ESE3-input in uno schema ESE3-modificato e si modifichi quest'ultimo secondo le Specifiche aggiuntive;
- 3.2) [1 p.] Si copi lo schema modificato in uno schema ESE3-tradotto. Mediante il comando Transform/Quick SQL, si traduca la parte di schema specificata, modificando lo script SQL in modo da essere compatibile con DB2 e permettere l'esecuzione del punto successivo, ed eventualmente aggiungendo quanto richiesto dalle Specifiche aggiuntive;
- 3.3) [3 p.] Si scriva l'istruzione SQL che modifica il DB come da specifiche (usare valori a scelta) e si definiscano i trigger necessari.
- NB: oltre a quanto richiesto, non è necessario definire altri trigger che garantiscano l'integrità dei dati.

4) Progettazione logica (6 punti totali)

Consegnare il file **ESE4.sql**

Dato lo schema concettuale in figura, disponibile anche nel file ESE4.lun, e considerando che:

- a) la gerarchia viene tradotta operando un collasso verso il basso;
- b) l'associazione R1 non viene tradotta separatamente;
- c) un'istanza di E4 non è mai associata a un'istanza di E1 che è istanza sia di E2 che di E3;



- 4.1) [3 p.] Si progettino gli opportuni schemi relazionali e si definiscano tali schemi mediante uno script SQL compatibile con DB2;
- 4.2) [3 p.] Per i vincoli non esprimibili a livello di schema si predispongano opportuni trigger che evitino **inserimenti di singole tuple non corrette**.

NB: Per il punto 4.2), se necessario, si specifichino usando commenti SQL eventuali inserimenti di tipo transazionale (ossia, più INSERT nella stessa transazione)