

Matricola	Nome e Cognome	Valutazione

Si consideri il seguente schema relazionale:

ImpresaEdile(id, ragione_sociale, id_titolare)

Lavoratore(id, nome, cognome, impresa)

Cliente(id, nome, cognome, immobile)

Cantiere(id, idAzienda, idImmobile, data_inizio, data_fine)

LavoratoriCantiere(idCantiere, idLavoratore, ruolo, ore_totali)

1. Identificare le chiavi primarie ed esterne dello schema [0 corrette, -1 errate]

2. Algebra

- a. Identificare le imprese che non hanno nessun cantiere [3 punti].

$PROJ_id(Impresa) - RID_{\{idAzienda \rightarrow id\}}(PROJ_idAzienda(Cantiere))$

- b. Per ogni cantiere identificare i lavoratori che hanno lavorato per più ore [4 punti].

$$C_1 := \sigma_{id_cantiere \rightarrow id_C, (LavoratoriCantiere)} \\ id_lavoratore \rightarrow id_L, \\ ruolo \rightarrow r_l, \\ ore_totali \rightarrow ore_T$$

$$C_2 := \pi_{id_cantiere, ore_totali, id_lavoratore, ruolo} \left(\sigma_{\begin{matrix} id_cantiere = id_C \wedge \\ ore_totali < ore_T \end{matrix}} (LavoratoriCantiere \times C_1) \right)$$

$$\pi_{id_cantiere, id_lavoratore} (LavoratoriCantiere - C_2)$$

Matricola	Nome e Cognome	Valutazione

3. SQL

- a. Identificare i clienti che hanno avuto il maggior numero di cantieri, indicare anche il numero complessivo dei lavoratori per tutti i cantieri di quel cliente [5 punti].

```
SELECT count(DISTINCT ca.id) x, count(lc.idLavoratore), c.id
FROM Cantiere ca, Cliente c, LavoratoriCantiere lc
WHERE c.immobile=ca.idImmobile AND lc.idCantiere = ca.id
GROUP BY c.id
HAVING x >= ALL (SELECT count(DISTINCT ca.id)
FROM Cantiere ca, Cliente c, WHERE c.immobile=ca.idImmobile
GROUP BY c.id)
```

- b. Trovare i cantieri che sono stati aperti per più tempo, per questi indicare anche il numero di ore complessive di tutti i lavoratori impiegati in tale cantiere [6 punti].

```
SELECT cantiere.id, sum(oretotali)
FROM cantiere, lavoratoriCantiere
WHERE data_fine IS NOT NULL AND data_fine-data_inizio >= SELECT
MAX(data_fine-data_inizio) FROM cantiere WHERE data_fine IS NOT
NULL) ANDcantiere.id=lavoratoriCantiere.idCantiere
GROUP BY cantiere.id`
```

4. Trigger

Implementare un trigger che, quando la data di fine lavori di un cantiere è valorizzata, calcoli per ogni lavoratore in tale cantiere il numero di ore totali lavorate. Si consideri una giornata lavorativa standard di **8 ore**. Ignorare la possibilità di avere domeniche e festivi [5 punti].

```
CREATE TRIGGER T1
AFTER UPDATE OF data_fine on CANTIERE
FOR EACH ROW
UPDATE lavoratoriCantiere
SET oretotali=(new.data_fine-new.data_inizio)*8
WHERE idCantiere=new.id
```

- a. Identificare i clienti che hanno avuto il maggior numero di cantieri, indicare anche il numero complessivo dei lavoratori per tutti i cantieri di quel cliente [5 punti].

```
SELECT CL.id, count(distinct CA.id) conteggio, count(distinct LC.id) lavoratori
FROM Cliente CL, Cantiere CA, LavoratoriCantiere LC
WHERE CL.immobile = CA.immobile
      AND CA.id = LC.idcantiere
GROUP BY CL.id
HAVING conteggio >= ALL (
    SELECT count(distinct cas.id)
    FROM cantiere CA1, cliente CL1
    WHERE CL1.immobile = CA1.immobile
    GROUP BY CL1.id
)
```

Trovare i cantieri che sono stati aperti per più tempo, per questi indicare anche il numero di ore complessive di tutti i lavoratori impiegati in tale cantiere [6 punti].

SELECT CA.id, sum (LC.oretotali)

FROM Cantiere CA, LavoratoriCantieri LC

WHERE LC.idcantiere = CA.id

AND (CA.data_fine - CA.data_inizio) = (SELECT max (CA1.data_fine - CA1.data_inizio)
FROM Cantiere CA1)

GROUP BY CA.id

ImpresaEdile(id, ragione sociale, id titolare)

Lavoratore(id, nome, cognome, impresa)

Ciente(id, nome, cognome, immobile)

Cantiere(id, idAzienda, idImmobile, data_inizio, data_fine)

LavoratoriCantiere(idCantiere, idLavoratore, ruolo, oretotali)