17-10-2022

Esercizio della volta scorsa.

Trovare le coppie di persone che hanno partecipato sempre alle stesse escursioni

Partecipante = PA

$$P = P_1 = Partecipante$$

si può "ridenominare" in IDP = IdPartecipante, IDE = IdEscursione

 $R_1 = \pi_{P.idp,P1.idp,P.ide}(P \bowtie_{P.idp>P1.idp,P.ide=P1.ide} P1)$ hanno partecipato ad almeno una escursione insieme

Uno dei due ha saltato un'escursione, sono quelli che dobbiamo rimuovere.

$$R_2 = (P \bowtie_{P.idp>P_1.idp,P.ide <> P_1.ide} P_1)$$

 $\pi_{P.idv,P1.idv,P.ide}(R_2)$ Escursioni dove P1 ha partecipato ma FORSE P2 no.

$$R_3 = \pi_{P.idp,P1.idp,P.ide}(R_2)/R_1$$

 $R_4=\pi_{P.idp,P_1.idp,P_1.ide}(R2)$ P1 ha partecipato ma P2 potrebbe non aver partecipato

$$R_4 = (\pi_{P.idp, P_1.idp, P_1.ide}(R2))/R_1$$

$$\pi_{P.idp,P_1.idp}(R1)/\pi_{P.idp,P1.idp}(R_3UR_4)$$

Dire ogni accompagnatore quante escursioni ha guidato

Non è possibile contare in algebra relazionale quindi questa query non si può risolvere.

Dovrei fare un numero indefinito di join per trovare la soluzione.

Esercizio 3.8

Si consideri lo schema relazionale composto dalle seguenti relazioni:

PROFESSORI (<u>Codice</u>, Cognome, Nome)
CORSI (<u>Codice</u>, Denominazione, Professore)
STUDENTI (<u>Matricola</u>, Cognome, Nome)
ESAMI (<u>Studente</u>, <u>Corso</u>, Data, Voto)

Formulare, con riferimento a tale schema, le espressioni dell'algebra, del calcolo relazionale su tuple e del Datalog, che producano:

- 1. Gli esami superati dallo studente Pico della Mirandola (supposto unico), con indicazione, per ciascuno, della denominazione del corso, del voto e del cognome del professore;
- 2. i professori che tengono due corsi (e non più di due), con indicazione di cognome e nome del professore e denominazione dei due corsi.

SOLUZIONE

http://www.star.dist.unige.it/~enrico/BasiDiDati/Esercizi/6008-5 03.pdf

SQL

Con SQL posso fare definizioni del modello di database e interrogazione del database.

SELECT = Prodotto cartesiano, selezione, ridenominazione, proiezione in algebra relazionale.

```
SELECT [DISTINCT] Attributi

FROM Tabelle
[WHERE Condizione]
```

Elenco di attributi, separati da virgole, proiezione From = come se facessi il prod. cart. fra le tabelle elencate

DISTINCT serve per eliminare i duplicati.

JLLL

- La query:
 - Considera il prodotto cartesiano tra le tabelle in Tabelle
 - 2. Fra queste seleziona solo le righe che soddisfano la *Condizione*
 - Infine valuta le espressioni specificate nella target list Attributi
- La SELECT implementa gli operatori Ridenominazione Proiezione, Selezione e Join dell'algebra relazionale
 - Più altro che vedremo più avanti

```
SELECT [DISTINCT] Attributi
FROM Tabelle
[WHERE Condizione]

Attributi ::= * | Attributo {, Attributo}
Tabelle ::= Tabella {, Tabella}
```

 Dove Tabella sta per una determinata relazione ed Attributo è uno degli attributi delle tabelle citate nella clausola FROM

^{*} Selezionatore di TUTTO

SELECT DISTINCT
$$A_1, A_2, ..., A_n$$

FROM $R_1, R_2, ..., R_m$
WHERE C

· Equivale a

$$\pi_{A_1,A_2,\dots,A_n}(\sigma_C(R_1\times R_2\times \dots \times R_m))$$

GUERL SOLONA LARELLA

STUDENTI

Nome	Matricola	Indirizzo	Telefono
Mario Rossi	123456	Via Etnea 1	222222
Ugo Bianchi	234567	Via Roma 2	333333
Teo Verdi	345678	Via Enna 3	444444

Vorrei conoscere indirizzo e telefono di Teo Verdi

SELECT *Indirizzo, Telefono*FROM Studenti
WHERE Nome='Teo Verdi'

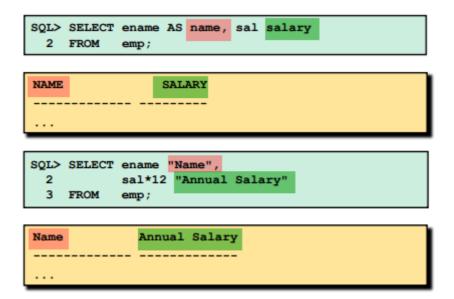
Indirizzo	Telefono
Via Enna 3	444444

SQL> SELECT ename, sal, sal+300
2 FROM emp;

ENAME	SAL	SAL+300
KING	5000	5300
BLAKE	2850	3150
CLARK	2450	2750
JONES	2975	3275
MARTIN	1250	1550
ALLEN	1600	1900
14 rows sele	cted.	

La precedenza degli operatori è quella classica. Prima la moltiplicazione/divisione e poi addizione/sottrazione. Le operazioni sono svolte solo su numeri perchè su altri tipi non ha senso.

Alias usato con la keyword AS oppure senza AS



Gli apici sono facoltativi per sostantivi singoli. E' obbligatorio quando ci sono caratteri speciali, tipo lo spazio fra 2 parole

```
SELECT p.Professore

FROM Corsi p, Esami e

WHERE p.Corso = e.Corso AND

Matricola = '123456'
```

- Professori con cui 123456 ha fatto esami
- Se i nomi degli attributi non sono univoci tra le tabelle si deve usare il nome della tabella nella select!
- E' consentito dall'uso della parola chiave DISTINCT nella clausola SELECT



Una relazione nell'algebra relazionale e' un insieme di tuple. In SQL i duplicati sono mantenuti.

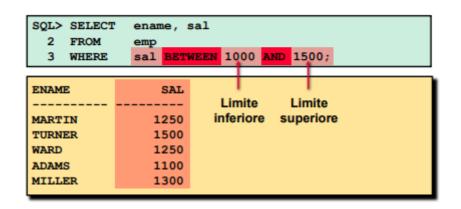
IMPIEGATI EMPNO ENAME JOB DEPTNO "...selezionare tutti gli impiegati 7839 KING PRESIDENT del dipartimeto 10" 7698 BLAKE MANAGER 30 7782 CLARK MANAGER 10 7566 JONES MANAGER 20 **IMPIEGATI** EMPNO ENAME 7839 KING PRESIDENT 10 7782 CLARK MANAGER 10 7934 MILLER CLERK 10

Predicati di confronto

Operatore	Significato
=	Uguale a
>	più grande di
>=	maggiore o uguale di
<	minore di
<=	minore o uguale a
<>	diverso

Operatori	Significato
BETWEENAND	compreso tra due valori
IN(list)	Corrisp. ad uno dei valori nella lista
LIKE	Operatore di pattern matching
IS NULL	Valore nullo

 BETWEEN consente la selezione di righe con attributi in un particolare range.



 E' usato per selezionare righe che hanno un attributo che assume valori contenuti in una lista.

```
SQL> SELECT empno, ename, sal, mgr
2 FROM emp
3 WHERE mgr IN (7902, 7566, 7788);
```

EMPNO	ENAME	SAL	MGR
7902	FORD	3000	7566
7369	SMITH	800	7902
7788	SCOTT	3000	7566
7876	ADAMS	1100	7788

- LIKE è usato per effettuare ricerche wildcard di una stringa di valori.
- Le condizioni di ricerca possono contenere sia letterali, caratteri o numeri.
 - · % denota zero o più caratteri.
 - _ denota un carattere.

```
SQL> SELECT ename
2 FROM emp
3 WHERE ename LIKE 'S%';
```

 Il pattern-matching di caratteri può essere combinato.

```
SQL> SELECT ename
2 FROM emp
3 WHERE ename LIKE '_A%';

ENAME
-----
MARTIN
JAMES
WARD
```

 l'identificatore ESCAPE (\) deve essere usato per cercare "%" o "_".

- L'operatore IS NULL controlla l'esistenza di valori null
- · Sintassi:

Espr IS [NOT] NULL

· Esempio:

SELECT Nome FROM Studenti WHERE Telefono IS NOT NULL

```
SQL> SELECT ename, mgr
2 FROM emp
3 WHERE mgr IS NULL;
```

[ENAME	MGR
ŀ		
ŀ	KING	

Operatore	Significato
AND	Restituisce TRUE if entrambe le condizioni sono TRUE
OR	Restituisce TRUE se <i>almeno</i> una delle condizioni è TRUE
NOT	Restituisce TRUE se la condizione è FALSE

```
SQL> SELECT empno, ename, job, sal
2 FROM emp
3 WHERE sal>=1100
4 AND job='CLERK';
```

EMPNO ENAME J	OB SAI	
	LERK 1100	

```
SQL> SELECT empno, ename, job, sal
2 FROM emp
3 WHERE sal>=1100
4 OR job='CLERK';
```

EMPNO	ENAME	JOB	SAL	
7839	KING	PRESIDENT	5000	
7698	BLAKE	MANAGER	2850	
7782	CLARK	MANAGER	2450	
7566	JONES	MANAGER	2975	
7654	MARTIN	SALESMAN	1250	
7900	JAMES	CLERK	950	
14 rows se	elected.			

```
SQL> SELECT ename, job
2 FROM emp
3 WHERE job NOT IN ('CLERK', 'MANAGER', 'ANALYST');
```

ENAME	JOB
KING	PRESIDENT
MARTIN	SALESMAN
ALLEN	SALESMAN
TURNER	SALESMAN
WARD	SALESMAN