

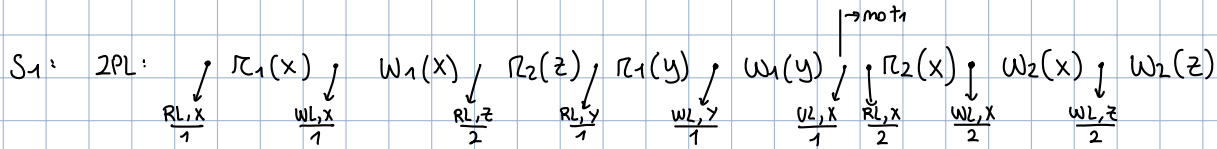
Esercizio 4:

Stabilire se i seguenti schedule sono o meno serializzabili rispetto al metodo del locking a due fasi, al metodo del locking a due fasi stretto e al metodo basato sui timestamp.

$s_1: r_1(x), w_1(x), r_2(z), r_1(y), w_1(y), r_2(x), w_2(x), w_2(z);$

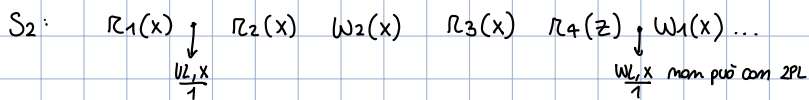
$s_2: r_1(x), r_2(x), w_2(x), r_3(x), r_4(z), w_1(x), w_3(y), w_3(x), w_1(y), w_5(x), w_1(z), w_5(y), r_5(z);$

$s_3: r_1(x), r_3(y), w_1(y), w_4(x), w_1(t), w_5(x), r_2(z), r_3(z), w_2(z), w_5(z), r_4(t), r_5(t).$



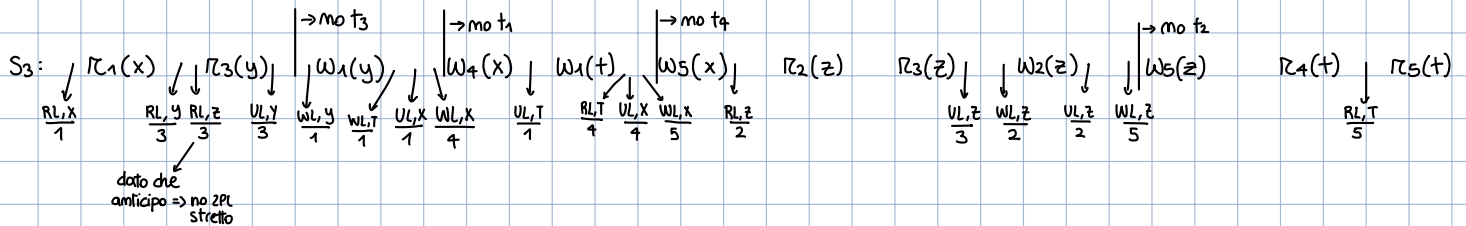
\bar{e} in 2PL e 2PL stretto

TIMESTAMP: $WTM(x) = 2$ $WTM(y) = 1$ $WTM(z) = 2$ $\Rightarrow \in$ TIMESTAMP
 $RTM(x) = 2$ $RTM(y) = 1$ $RTM(z) = 2$



$S_1 \notin$ 2PL, 2PL stretto

$S_1 \notin$ TIMESTAMP



$S_3 \in$ 2PL \notin 2PL perché anticipa i lock

$S_3 \notin$ TIMESTAP perché $(r_3(y), w_1(y))$

Esercizio 5:

Si consideri la seguente implementazione (incompleta) di una specializzazione parziale ed esclusiva:

```
create table Documento (  
    codice dom_codice primary key,  
    titolo dom_titolo not null  
);  
  
create table PDF (  
    codice dom_codice primary key references Documento,  
    pagine dom_pagine not null  
);  
  
create table Video (  
    codice dom_codice primary key references Documento,  
    durata dom_durata not null  
);
```

Si spieghi in che modo è possibile implementare il vincolo di esclusività in SQL, descrivendo anche il codice corrispondente.

Si utilizzano due TRIGGER che verificano all'inserimento/modifica che nell'altra tabella figlio non esista già un valore corrispondente

controllo nei PDF:

```
create or replace function controlla_pdf()  
returns trigger language plpgsql as $$  
declare  
    cod dom_codice;  
begin  
    select codice into cod from PDF where new.codice = codice; --new e' un oggetto Video  
    if not found then  
        return new;  
    else  
        raise exception 'C'è già un PDF con lo stesso codice';  
        return null;  
    end if;  
end;  
$$;
```

```
create trigger controllaPDF  
before insert or update on Video  
for each row  
execute procedure controlla_pdf();
```

```
create or replace function controlla_video()  
returns trigger language plpgsql as $$  
declare  
    cod dom_codice;  
begin  
    select codice into cod from Video where codice = new.codice;  
    if not found then  
        return new;  
    else  
        return null;  
    end if;  
end;  
$$;
```

```
create trigger controllaVideo  
before insert or update on PDF  
for each row  
execute procedure controlla_video();
```

create or replace function controllo_pdf()

returns trigger language plpgsql as \$\$

declare

cod dom_codice;

begin

select codice into dom_codice from PDF where new.codice = codice

if not found then

return new; //inserimento corretto

else

raise exception 'C'è già un PDF con questo codice';

return null;

end if;

end;

\$\$;

create trigger controllaPDF

after update or insert on Video

for each row

execute procedure controllo_pdf();

analogo per i DVD