

# **Livelli di isolamento Progetto**

- Studiare e sperimentare il comportamento di uno o più DBMS (si consiglia Postgres) nella gestione dei livelli di isolamento, con riferimento alle principali anomalie. Si suggerisce di fare riferimento anche ad esempi simili a quelli utilizzati nei lucidi seguenti (considerarli tutti, perché potrebbero essere trattati nelle prove d'esame o in quelle intermedie).
- Dal punto di vista realizzativo, si può procedere sviluppando applicazioni in Java con JDBC, oppure attraverso stored procedure oppure anche utilizzando gli ambienti interattivi. Consegnare una breve relazione che illustri i test fatti, mostrando i risultati dei test stessi (file di output o schermate o anche descrizione delle schermate).

# Postgres

- Si possono scrivere applicazioni, ma la soluzione più semplice consiste nell'utilizzare due client, alternando le operazioni (è importante documentare le operazioni)
- Utilizzare operazioni di lettura (select) e scrittura (insert o update)
- Si possono usare in scrittura semplici costanti (perché il controllo di concorrenza non usa la semantica delle operazioni), ma per capire meglio si possono usare
  - SELECT INTO per leggere (usando tabelle di appoggio)
  - INSERT o UPDATE con sottorinterrogazioni
- Ad esempio

```
create table conti (numero integer primary key, saldo integer);
insert into conti values (1,1000);
select saldo into appoggio from conti where numero=1;
update conti set saldo = (select saldo+100 from appoggio)
where numero = 1
```

## Esempio 1 (perdita di aggiornamento)

Start Transaction

isolation level XXX

Read(x)

$x=x+100$

Write(x)

Commit

Start Transaction

isolation level XXX

Read(x)

$x=x+1$

Write(x)

Commit

## Esempio 2 (esame 09/2012)

Start Transaction

Read(x)

$x=x+100$

Write(x)

Commit

Start Transaction

Read(x)

$x=x+1$

Write(x)

Commit

## Esempio 3

Start Transaction

Read(x)

$x = x + 100$

Write(x)

Abort

Start Transaction

Read(x)

$x = x + 1$

Write(x)

Commit

## Esempio 4, lettura inconsistente

Start Transaction

Read(x)

Start Transaction

Read(x)

$x=x+20$

Write(x)

Commit

Read(x)

Commit

# Repeatable read vs serializable

start transaction isolation level ...

```
select count(*)  
from studenti
```

```
insert into studenti values (...)  
commit
```

start transaction isolation level ...

```
select count(*)  
from studenti;  
insert into studenti values ( ... );
```

```
commit
```

# Provare anche i vincoli di integrità (chiave)

```
start transaction  
isolation level ...;  
insert into conti  
values (2,1000)
```

```
commit
```

```
start transaction  
isolation level ...;  
insert into conti  
values (2, 1000)
```

```
commit
```

# Ancora

- Ripetere la prova precedente con due valori diversi della chiave
- E poi con letture (ad esempio `SELECT *`) in ciascuna delle due transazioni, prima del commit