



Commitment control

for everyone

MK1

Marco Riva
www.markonetools.it

IBMCHAMPION 2021-2025 Power area

Ultimo aggiornamento: 01/02/2026

IBM i

RPG SQL

1



KEY POINTS

- Critical
- Important
- Nice to have

Gruppi attivazione
Ambito
Attivazione
Operazioni implicate

Livello isolamento
Concorrenza accesso
Monitoraggio

Deadlock
Savepoint
Soft commit
Giornali

Commitment control for everyone

2



3



4



5

Controllo di sincronia

Concetti base

Commitment control for everyone

5



6

Controllo sincronia (commitment control)



- consente di **definire e consolidare** un gruppo di modifiche sul database come **unità logica di lavoro** (LUW) = **transazione**



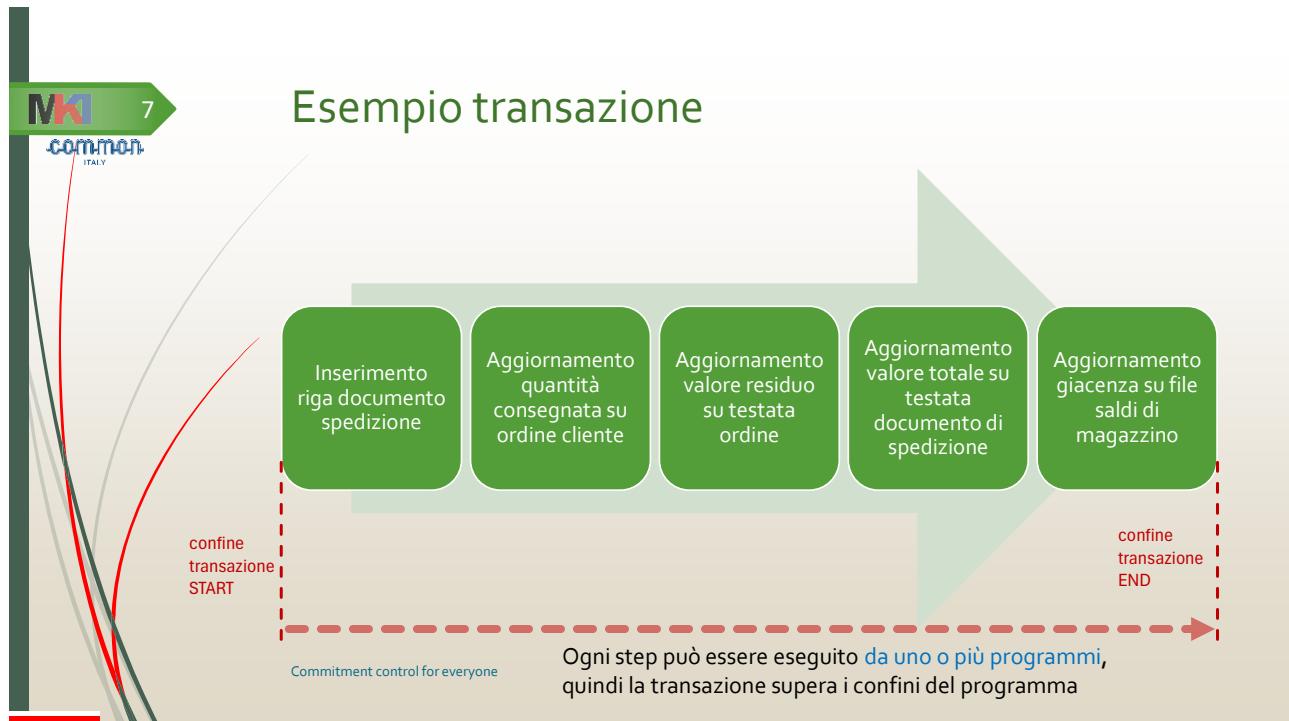
- **transazione**
- è un gruppo di modifiche a uno o più file di database che dal punto di vista dell'utente appaiono come una singola modifica



- **integrità e coerenza del dato**
- Assicura che l'intero gruppo di modifiche apportate in una transazione o vengano tutte consolidate sul db (*commit*) o tutte annullate (*rollback*)

Commitment control for everyone

6



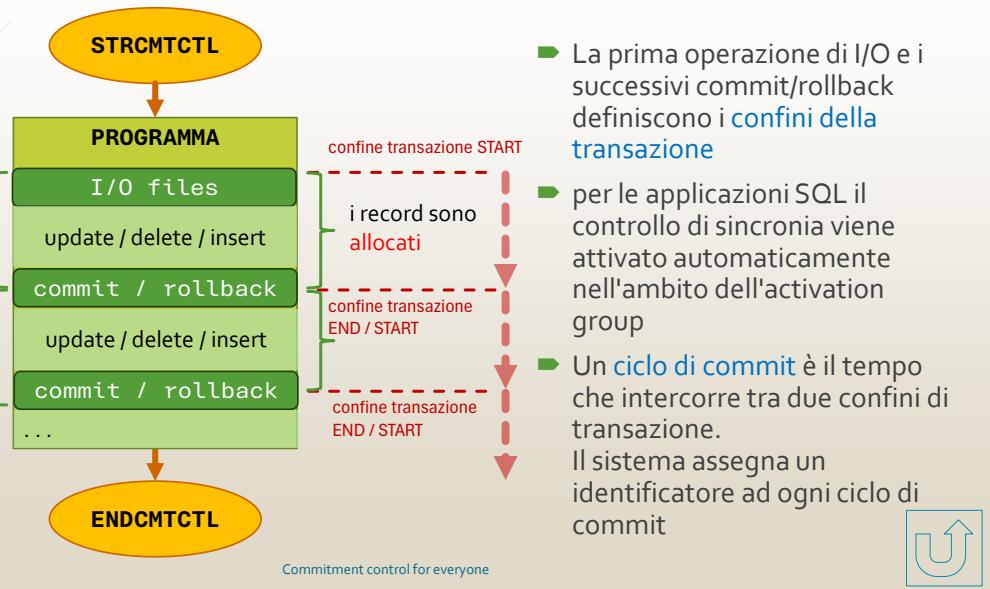
7



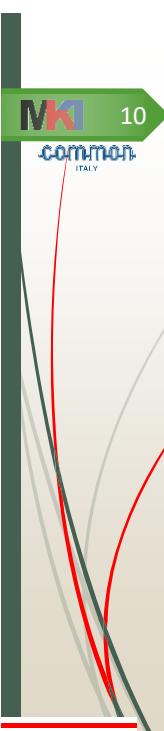
8



Controllo di sincronia – schema logico



9



Controllo sincronia: prerequisiti

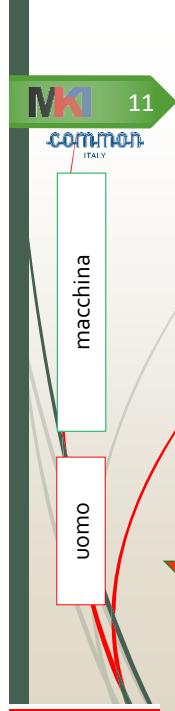
- ▶ I files devono essere registrati sullo **stesso giornale** con entrambe le immagini before e after.
Se il file è registrato solo con immagine after, il sistema registrerà anche l'immagine before
- ▶ **I/O nativo**: avviare definizione del controllo sincronia (**STRCMCTL**) e definire i file con keyword **COMMIT**
- ▶ **I/O SQL**: impostare il livello di isolamento (**set option commit**) diverso da ***NONE**

create schema o create collection
creano automaticamente una nuova libreria con il giornale di default QSQJRN e qualsiasi tabella creata nella libreria viene registrata sul giornale

Commitment control for everyone

Se si crea la libreria con **CRTLIB** occorre creare manualmente ricevitore e giornale ed attivare la registrazione con **STRJRNLIB**

10

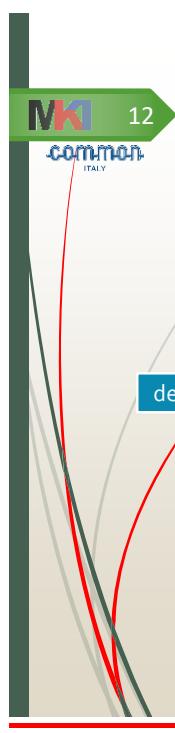


Perché lo si teme?

- ▶ richiede che tutti i file siano registrati sullo stesso giornale. Quindi si evita per timore di decadimento di performance e di occupazione spazio disco
- ▶ prima di RPG IV era macchinoso l'uso opzionale del controllo di sincronia nei programmi
- ▶ prima di ILE il controllo di sincronia agiva solo a livello di job e non dell'activation group
- ▶ poiché i file sono registrati su giornale richiede maggior attenzione nella manipolazione di oggetti e librerie
- ▶ **complessità del disegno applicativo per definire correttamente le transazioni e gli activation group**

Commitment control for everyone

11



Ambito delle transazioni

- ▶ L'ambito (=scope) delle transazioni può essere a livello
 - ▶ **Lavoro:** ogni programma chiamato dopo STRCMTCTL CMTSCOPE (*JOB) in esecuzione in qualsiasi gruppo di attivazione che non abbia un controllo di sincronia specifico del proprio gruppo di attivazione userà il controllo di sincronia a livello di lavoro
 - ▶ **Gruppo di attivazione:** STRCMTCTL CMTSCOPE (*ACTGRP) oppure applicazione SQL con SET OPTION COMMIT diverso da *NONE. Solo i programmi in esecuzione nel gruppo di attivazione useranno il controllo di sincronia specifico di quel gruppo
 - ▶ **Transazione:** avviati con XA APIs for Transaction Scoped Locks. Questa API è utilizzata per associare la definizione del controllo sincronia a uno specific thread o a una connessione SQL e non all'activation group

Commitment control for everyone



12

Job e activation group

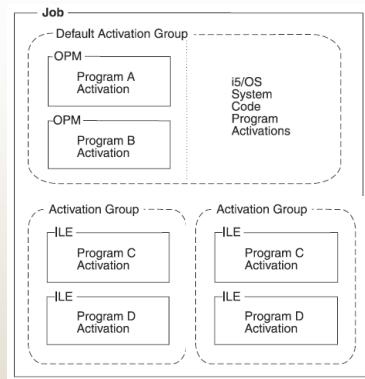
Un breve ripasso

Commitment control for everyone

13

Job e activation group: program isolation

- ▶ L'**activation group** contiene le **risorse** necessarie ad eseguire il programma. Le risorse sono:
 - ▶ Static program variables
 - ▶ Dynamic storage
 - ▶ Temporary data management resources
 - ▶ ODP (Open Data Path)
 - ▶ **Commitment definitions**
 - ▶ SQL cursors
 - ▶ HFS (Hierarchical File System)
- ▶ All'avvio di un job, *automaticamente* vengono creati 2 activation group di **default** usati da tutti i programmi OPM e ILE OPM compatible



Fonte IBM

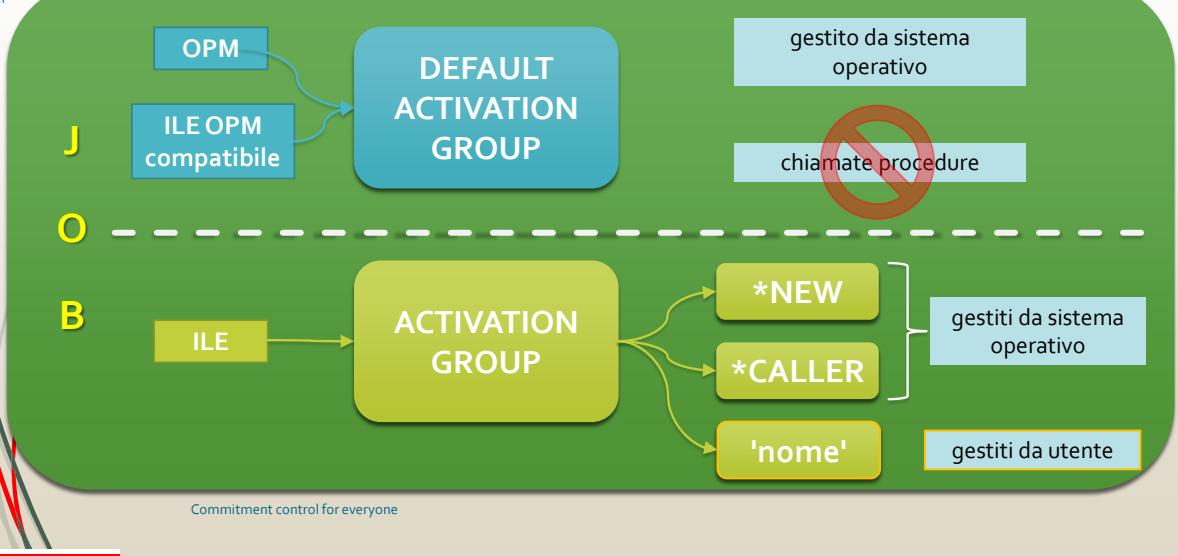
Commitment control for everyone

14

MK1
common
ITALY

15

Activation group in sintesi



15

MK1
common
ITALY

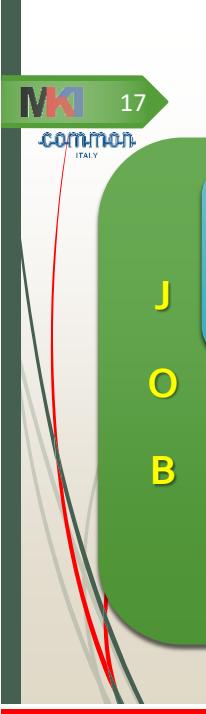
16

Activation group: comandi principali

- **RCLACTGRP**: libera le risorse di un AG nominale
- **CRTSQLRPGI** parametro **CLOSQLCSR**: determina quando viene chiuso automaticamente un cursore SQL. *ENDACTGRP o *ENMOD.
- **OVR...** parametro **OVRSCOPE**: determina l'ambito di validità della sostituzione. *ACTGRPDFN o *CALLLVL o *JOB.
- **DLTOVR** parametro **LVL**: determina quale livello di sostituzioni eliminare. *ACTGRPDFN o * o *JOB.

Commitment control for everyone

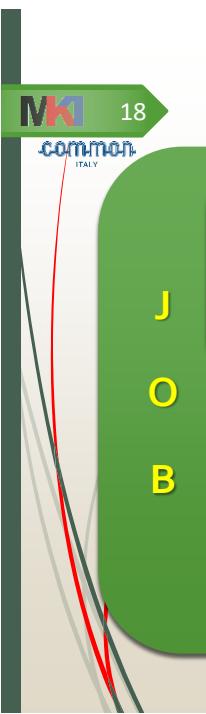
16

**Comm.lt. quiz****DEFAULT ACTIVATION GROUP****STRCMTCTL *JOB****PGMC → PGMD****ACTIVATION GROUP****XYZ****PGMA**

17



PGMA condivide la stessa definizione di controllo di sincronia del programma PGMC e PGMD?

**Comm.lt. quiz****DEFAULT ACTIVATION GROUP****PGME***embedded sql
option commit = *chg***ACTIVATION GROUP 23****PGMF***embedded sql
option commit = *chg*

18



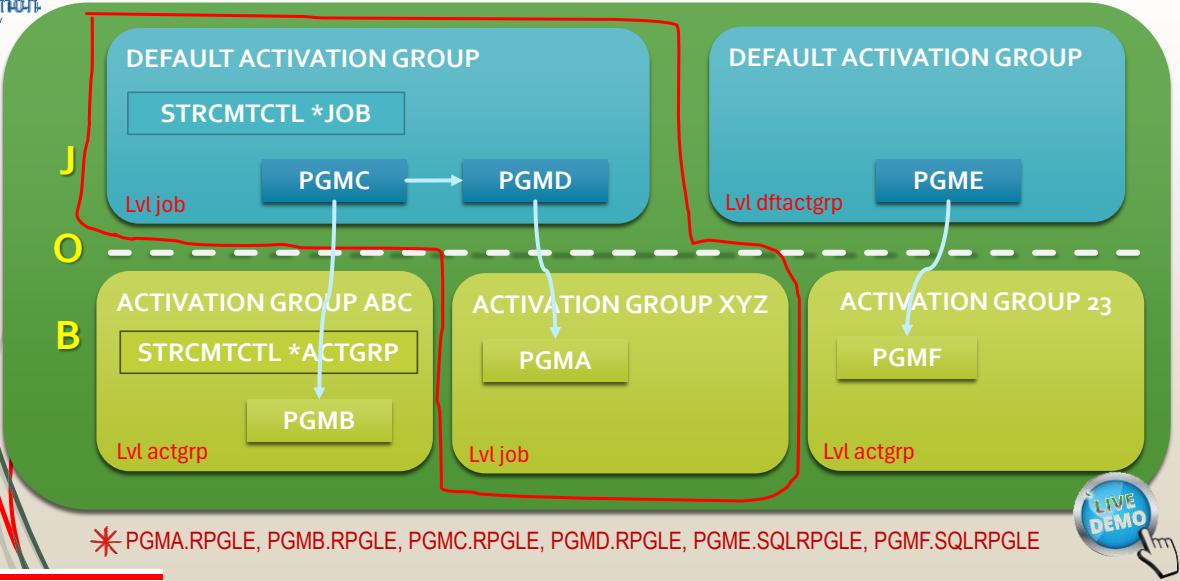
PGMF condivide la stessa definizione di controllo di sincronia del programma PGME?



MK1
common
ITALY

19

Controllo sincronia - isolamento transazioni/1



19

MK1
common
ITALY

20

Controllo sincronia - isolamento transazioni/2

Annotazioni allo schema precedente

- PGMC avvia STRCMTCTL a livello *JOB
- i programmi PGMD e PGMA chiamati da PGMC ereditano la definizione di commit a livello job
- il programma PGMB avvia STRCMTCTL a livello *ACTGRP quindi viene creata una definizione di commit a livello dell'activation group ABC
- il PGME (embedded SQL) nel dftactgrp non avvia esplicitamente STRCMTCTL quindi si crea una definizione implicita di commit a livello *DFTACTGRP
- il PGMF (embedded SQL) chiamato da PGME apre un nuovo activation group e implicitamente crea una nuova definizione di commit a livello dell'activation group

Commitment control for everyone

20



Demo - CALL PGMC/1

WRKCMTDFN

DSPLY Avviato contr.sincr. *JOB. Invio per proseguire

Opz	Definiz.	N°	Uten.	Nome	Sottopr.	Utente
■	*JOB	468667	MRIVA	MRIVAA	*NONE	MRIVA

Opz	Definiz.	N°	lav.	ID LUW	Stato
■	*JOB	468667	APPN.████████.X'BCFEBB5E077B'.00001	RIPRISTINO	

Visualizzazione stato definizione commit 28/01/26 12:45:29

Lavoro:	Utente:	Numero:
██████A	██████	468667

Sottopr. : *NONE
ID LUW (Logical Unit of Work) : APPN.████████.X'BCFEBB5E077B'.00001
ID spazio blocco : UDB_01000000000409ED
Definizione commit : *JOB
Gruppo attivazione :
Gruppo ASP : *SYSBAS
Ubicazione risorsa : NESSUNO
Livello blocco predefinito : *CHG
Utente :
Modifiche in sospeso locali : NO
Ruolo :
Stato : RIPRISTINO
Registrazione data/ora :

Risincronizz. in corso : NO
Operaz. euristica :
Lavoro attivo : SI
Lavoro server :
Ambito blocco : *JOB
Supero tempo transazione :
Attesa massima blocco :

Commitment control for everyone

21



Demo - CALL PGMC/2

WRKCMTDFN

DSPLY Eseguito update 000200. Proseguire? (S/N)

Opz	Definiz.	N°	Risinc.	Modif.
■	*JOB	468667	in corso	locali sospese

Visualizzazione stato definizione commit 28/01/26 12:48:19

Lavoro:	Utente:	Numero:
██████A	██████	468667

Sottopr. : *NONE
ID LUW (Logical Unit of Work) : APPN.████████.X'BCFEBB5E077B'.00001
ID spazio blocco : UDB_01000000000409ED
Definizione commit : *JOB
Gruppo attivazione :
Gruppo ASP : *SYSBAS
Ubicazione risorsa : LOCALE
Livello blocco predefinito : *CHG
Utente :
Modifiche in sospeso locali : SI

Visualizzazione dettagli voce di giornale INO 6 12:46:46

Giornale	Libreria
QSQJRN	MK1SAMPLE

Sequenza : 6729
Codice : R - Operazione su un record specifico
Tipo : UP - Aggiornamento, immagine-successiva

Oggetto : EMPLOYEE
Libreria : MK1SAMPLE
Membro : EMPLOYEE
Tipo : *ODDS
Data : 28/01/26
Ora : 12:46:46,837008
Contrassegno : 0
Conteggio/RRN : 18
ID ciclo di commit : 6727
Visualizzazione delle voci di giornale

le : QSQJRN Libreria : MK1SAMPLE
di sequenza più grande su questa schermata . : 0000000000000000
ere le opzioni e premere Invio.
visualizzazione intera voce

Sequenza Cod. Tipo Oggetto Libreria Lavoro Or.
6727 C SC start ciclo di commit
6728 R UB EMPLOYEE MK1SAMPLE 12
6729 R UP EMPLOYEE MK1SAMPLE 12

22

MK1 23

common
ITALY

PGMC

Demo - CALL PGMC/3

WRKCMTDFN

DSPLY Avvio programma PGMD. Invio per proseguire.

Opz	Definiz.	N°	Uten.	Nome	Sottopr.	Utente
*JOB	commit	lav.	lav.	lav.	*NONE	MRIVA

Visualizzazione stack di chiamata						
Lavoro:	Utente:	Numero:	-----Gruppo attivazione-----			
Sottoproc.: 0000016A						
			Tipo	Programma	Nome	Numero
1	QUIMNDRV	QSYS	QCMD	QSYS	*DFTACTGRP	0000000000000001
2	QUIMGFLW	QSYS	QUICMENU	QSYS	*DFTACTGRP	0000000000000001
3	QUICMD	QSYS	2	QUIMNDRV	*DFTACTGRP	0000000000000001
	PGMC	MK1SQL	3	QUIGFLW	*DFTACTGRP	0000000000000001
	PGMC	MK1SQL	→	QUICMD	*DFTACTGRP	0000000000000002
	PGMD	MK1SQL		PGMC	*DFTACTGRP	0000000000000002
	PGMD	MK1SQL		PGMD	*DFTACTGRP	0000000000000002

Commitment control for everyone

23

MK1 24

common
ITALY

PGMC

Demo - CALL PGMC/4

WRKCMTDFN

DSPLY Eseguito update 000210. Proseguire? (S/N)

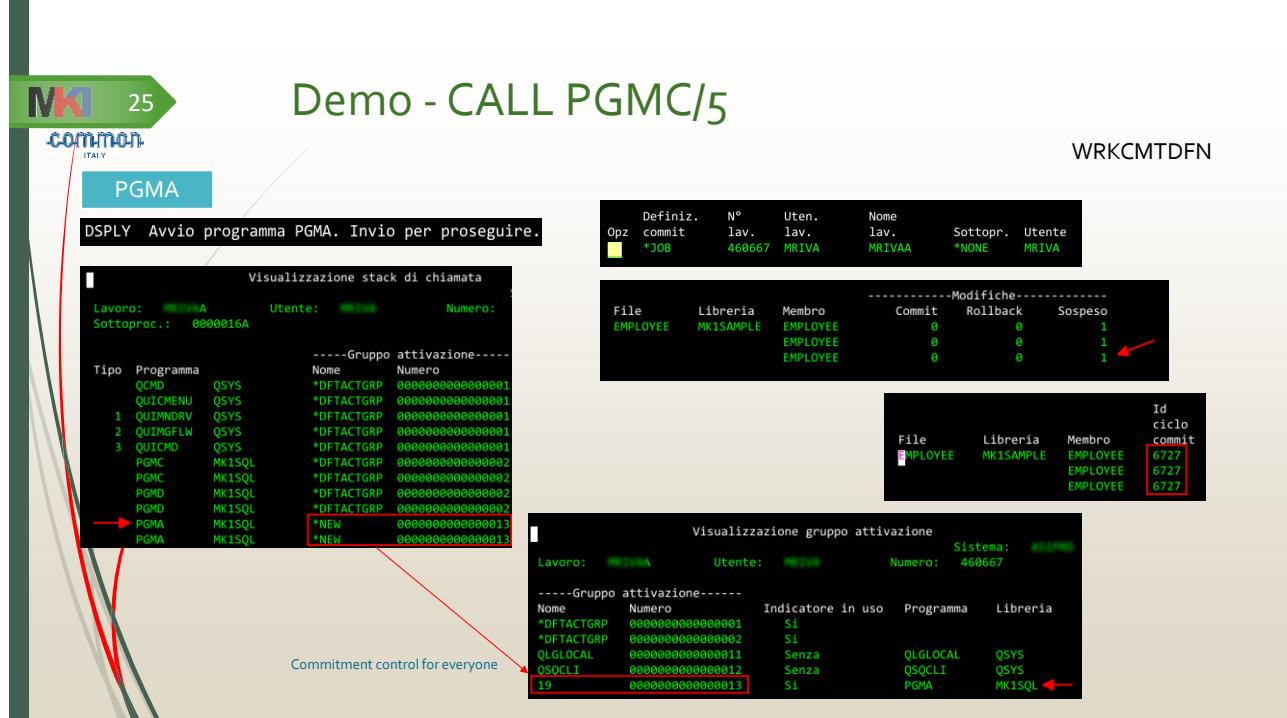
Visualizz. stato livello record						
Lavoro:	Utente:	Numero:	Sistema: 460667			
Definizione commit : *JOB						
File	Libreria	Membro	Modifiche			
EMPLOYEE	MK1SAMPLE	EMPLOYEE	Commit	Rollback	Sospeso	1
		EMPLOYEE	0	0	0	1

File	Libreria	Membro	Id	ciclo
EMPLOYEE	MK1SAMPLE	EMPLOYEE	6727	6727

Opz	Giornale	Libr.	Id	ciclo	Blocchi	Modifiche
QSQJRN	MK1SAMPLE	6727			2	2

Commitment control for everyone

24



25

Demo - CALL PGMC/6



26

MK1 27

Demo - CALL PGMC/7

WRKCMTDFN

PGMB

DSPLY Eseguito update 000240. Proseguire? (S/N)

Opz	Definiz.	N°	Risinc.	Modif.
	commit	lav.	in corso	locali sospese
*JOB		460667	NO	SI
20		460667	NO	SI

Visualizz. stato livello record Sistema: ALM000

Lavoro: **20** Utente: **460667** Numero: **460667**

Definizione commit : **20**

File	Libreria	Membro	-----Modifiche-----
EMPLOYEE	MK1SAMPLE	EMPLOYEE	Commit 0 Rollback 0 Sospeso 1
Opz	Giornale	Libr.	Id ciclo commit
QSORN	MK1SAMPLE		6735

BloCCI di record Modifiche in sospeso

Definizione commit : **20**

File	Libreria	Membro	Id ciclo commit
EMPLOYEE	MK1SAMPLE	EMPLOYEE	6735

Sequenza	Cod.	Tipo	Oggetto	Libreria	Lavoro	Ora
6735	C	SC	start ciclo di commit			15:25:09
6736	R	UB	EMPLOYEE	MK1SAMPLE		15:25:09
6737	R	UP	EMPLOYEE	MK1SAMPLE		15:25:09

Commitment control for everyone

27

MK1 28

Demo - CALL PGMC/8

WRKCMTDFN

PGMB

DSPLY PGMB ha eseguito commit. Invio per proseguire

Opz	Definiz.	N°	Risinc.	Modif.
	commit	lav.	in corso	locali sospese
*JOB		460667	NO	SI
20		460667	NO	NO

File	Libreria	Membro	-----Modifiche-----
EMPLOYEE	MK1SAMPLE	EMPLOYEE	Commit 1 Rollback 0 Sospeso 0

Visualizzazione stato definizione commit 20/01/26 15:31:25

Lavoro: **20** Utente: **460667** Numero: **460667**

Sottopr. : *NONE
ID LUW (Logical Unit of Work) . . . : APPN. **X'BD21C5AF077B'**.00002
ID spazio blocco : UDB_01000000000409F5
Definizione commit : **20**
Gruppo attivazione : **20**
Gruppo ASP : *SYSBAS

Limite blocco : 500000000
Commit durevole : SI
Numero di commit : **1**
Numero di rollback : 0
giornale predefinito : Libreria :

Commitment control for everyone

28

MK1
common
ITALY

29

Demo - CALL PGMC/9

WRKCMTDFN

PGMB

DSPLY PGMB arresterà il contr.sincr. Invio per proseguire
 DSPLY Arresto controllo di sincronia con errori
 ENDCMTCTL non è consentito. Sono presenti delle modifiche in sospeso.

DSPLY PGMB ha chiuso il file EMPLOYEE. Invio per proseguire

DSPLY PGMB arresterà il contr.sincr. Invio per proseguire

Commitment control for everyone

Visualizz. stato livello record Sistema: ASIPRD
 Lavoro: MRIVA Utente: MRIVA Numero: 460667
 Definizione commit : 28
 File Libreria Membro Liv blocco Stato Risoluzione
 EMPLOYEE MKISAMPLE EMPLOYEE *CHG APERTO accessi contemp.
 *WAIT

Visualizz. stato livello record Sistema: ASIPRD
 Lavoro: MRIVA Utente: MRIVA Numero: 460667
 Definizione commit : 28
 File Libreria Membro Liv blocco Stato Risoluzione
 accessi contemp.
 (Non ci sono modifiche a livello record sotto controllo di commit.)

Definiz.	N°	Uten.	Nome	Sottopr.	Utente
Opz commit	lav.	lav.	lav.	*NONE	MRIVA
*JOB	460667	MRIVA	MRIVAA		

29

MK1
common
ITALY

PGMC

Demo - CALL PGMC/10

WRKCMTDFN

DSPLY PGMC arresterà il contr.sincr. Invio per proseguire
 DSPLY Arresto controllo di sincronia con errori

Chiusura PGMC

SIGNOFF

Visualizz. stato livello record Sistema: ASIPRD
 Lavoro: MRIVA Utente: MRIVA Numero: 460667
 Definizione commit : *JOB
 File Libreria Membro Liv blocco Stato Risoluzione
 EMPLOYEE MKISAMPLE EMPLOYEE *CHG APERTO accessi contemp.
 *WAIT
 CHIUSO

Definiz.	N°	Uten.	Nome	Sottopr.	Utente
Opz commit	lav.	lav.	lav.	*NONE	MRIVA
*JOB	460667	MRIVA	MRIVAA		

(Non ci sono definizioni di commit attive)

CPF8356 Diagnostica 30 20/01/26 15:41:04,336486 QTNEND QSYS 2075 QWTPITP2 QSYS 0439
 Messaggio . . . : Il controllo di commit è terminato con 3 modifiche non sottoposte a commit.
 5770SSI V7R5M0 220415 Registrazione lavoro ASIPRD 20/01/26 15:41:04 CET Pag. 14
 Nome lavoro : MRIVA Utente : MRIVA Numero : 460667
 Descrizione lavoro : QPF2JOB Libreria : QGPL
 IDMSG TIPO GRAV DATA ORA DA PGM LIBERTA INST A PGM LIBRERIA INST
 Causa . . . : È stato eseguito il rollback di modifiche in sospeso poiché vi erano modifiche non sottoposte a commit nel momento in cui è terminato il controllo di commit per la definizione di commit *JOB con un id LWM (logical unit of work/unità logica di lavoro) APPN. X'BCFEBB5E077B'.00001.

30

Comm.lt. quiz

31

JOB

DEFAULT ACTIVATION GROUP

STRCMTCTL *JOB
PGMG

STRCMTCTL *ACTGRP
PGMH

* PGMG.RPGL, PGMH.RPGL

PGMH può aprire una definizione di controllo sincronia a livello *ACTGRP?

YES **NO**

3

31

32

Controllo sincronia - isolamento transazioni/3

Solo programmi in ambiente ILE possono aprire transazioni in un gruppo di attivazione diverso da quello di default

Ogni transazione può allocare al max 500.000.000 di record, ma si consiglia (*IBM documentation*) di non superare i 2.000 record per transazione

in un lavoro possono esistere più transazioni contemporanee e indipendenti se e solo se nel job sono in esecuzione uno o più programmi ILE

Commitment control for everyone

32

Controllo sincronia - isolamento transazioni/4

In un ambiente misto OPM e ILE se si desidera che tutti i programmi **condividano la medesima definizione** di controllo di sincronia è essenziale che l'ambito sia ***JOB**

Commitment control for everyone

33

Livelli di isolamento delle transazioni



integrità e sicurezza del dato

Livello	Descrizione	Attivo	Letture sporche	Record fantasma	Allocazione record letti	Note
*RR/RR	LETTURA RIPETIBILE (Repeatable Read or Serializable)	✓	✗	✗	✓	Garantisce che i record letti non possono essere modificati da un altro gruppo di attivazione e che ogni record modificato in un altro gruppo di attivazione non può essere letto fino a quando viene consolidato Allocata tutti i record letti e il gruppo di attivazione è completamente isolato rispetto ad altri
*ALL/RS	LETTURA STABILE (Read Stability)	✓	✗	✓	✓	Garantisce che i record letti non possono essere modificati da un altro gruppo di attivazione e che ogni record modificato in un altro gruppo di attivazione non può essere letto fino a quando viene consolidato → come RR Record "fantasma": a differenza di RR eseguendo più volte la stessa query possono comparire i record aggiunti in altri gruppi di attivazione
*CS/CS	LETTURA SINCRONIZZATA (Cursor Stability or Read Committed)	✓	✗	✓	✗	Garantisce che ogni record modificato in un altro gruppo di attivazione non può essere letto fino a quando viene consolidato → come RR o RS A differenza di RR e RS i record letti possono essere modificati da altri gruppi di attivazione
*CHG/UR	LETTURA NON SINCRONIZZATA (Uncommitted Read)	✓	✓	✓	✗	Livello più basso di controllo di sincronia. Consente di leggere record modificati in altri gruppi di attivazione anche se non ancora consolidati
*NONE/NC	NESSUNO	✗	✓	✓	✗	Ogni singolo aggiornamento è effettivamente sincronizzato quando viene completato. Non è possibile eseguire commit/rollback

Commitment control for everyone

<https://www.ibm.com/docs/en/i/7.6.0?topic=concepts-isolation-level>

34

Livelli di isolamento delle transazioni/2

- ▶ **Letture sporche:** è possibile leggere dati che sono stati variati ma non sincronizzati da un altro lavoro.
- ▶ **Record fantasma:** la LUW 1 legge un set di record che soddisfa certi criteri. La LUW 2 inserisce un nuovo record che soddisfa i criteri di selezione della query della LUW 1. La LUW 1 riesegue la query e legge anche i nuovi record
- ▶ **Lettura non ripetibile:** la LUW 1 legge un record. La LUW 2 modifica quel record e lo consolida. La LUW1 rilegge il medesimo record ed ottiene i nuovi dati consolidati dalla LUW 2

<https://www.ibm.com/docs/en/i/7.6.0?topic=level-comparison-isolation-levels>

Commitment control for everyone

35

Livello di allocazione

- ▶ Il parametro **LCKLVL** in STRCMTCTL determina il livello di allocazione dei record di default

Tipo	Descrizione	Conseguenze
* ALL	allocazione dei record modificati e letti durante tutta la transazione	- anche un record letto senza scopo di aggiornamento è allocato
* CS	allocazione dei record modificati durante tutta la transazione allocazione dei record letti fino al rilascio o alla successiva lettura	- un altro lavoro non può leggere record per aggiornamento che sono già stati letti dal lavoro corrente - il lavoro corrente non può leggere record per aggiornamento che sono stati allocati di tipo *update in un altro lavoro
* CHG	allocazione dei record modificati durante tutta la transazione	

Commitment control for everyone

<https://www.ibm.com/docs/en/i/7.6.0?topic=control-commit-lock-level>

36

Prerequisiti per attivare controllo di sincronia

Specifiche di controllo e definizione file

Opzioni precompilatore SQL

Commitment control for everyone

37

H specifiche di controllo

- ▶ Keyword per activation group
 - ▶ DFTACTGRP (*YES | *NO)
 - ▶ ACTGRP (*STGMDL | *NEW | *CALLER | '*nome*')

F specifiche definizione file

- ▶ keyword COMMIT
- ▶ keyword USROPN COMMIT (*ind-variable*)
se la variabile *ind-variable* è impostata a '1' il controllo di sincronia è attivato, se '0' invece non viene attivato

Commitment control for everyone

38



39

STRCMTCTL / ENDCMTCTL

► STRCMTCTL

avvia una definizione di controllo di sincronia

- ▶ Se già attivo → errore CPF8351

► ENDCMTCTL

arresta una definizione di controllo di sincronia

- ▶ Interattivo: se eseguito con modifiche in sospeso o con risorse aperte → CPA8350 o CPF8355
 - ▶ CM: esegue commit e prosegue
 - ▶ RB: esegue rollback e prosegue
- ▶ Batch: se eseguito con modifiche in sospeso o con risorse aperte → rollback

Commitment control for everyone

39



40

Embedded SQL

- ▶ Parametro **COMMIT** nei comandi di compilazione
- ▶ Oppure nel sorgente si può specificare la direttiva di compilazione con l'istruzione SQL set option. P.es.:


```
exec sql set option commit = *CHG;
```
- ▶ Oppure il livello di isolamento può essere specificato in ogni singola istruzione SQL DELETE, INSERT, SELECT, UPDATE
(*isolation-clause*)


```
... with ur;
```
- ▶ **Implicitamente** viene avviato il controllo di sincronia a livello di **activation group**

Commitment control for everyone

<https://www.ibm.com/docs/en/i/7.6.0?topic=statement-isolation-clause>

40

Chiudere una transazione/1

- ▶ Una transazione viene chiusa esplicitamente con le operazioni di commit (vengono consolidate le modifiche) o rollback (vengono annullate le modifiche)
- ▶ I codici operativi RPG per commit e rollback sono:
COMMIT, ROLBK
- ▶ Le istruzioni SQL per commit e rollback sono
`exec sql commit;`
`exec sql rollback;`

Commitment control for everyone

41

Chiudere una transazione/2

- ▶ **COMMIT**
 - ▶ Consolida tutte le modifiche eseguite sui record dal precedente commit/rollback
 - ▶ Rilascia tutti i lock sui record
 - ▶ Non viene alterata la posizione dei file
- ▶ **ROLLBACK**
 - ▶ Annulla tutte le modifiche eseguite sui record a partire dal precedente commit/rollback o dal savepoint (se embedded SQL)
 - ▶ Rilascia tutti i lock sui record
 - ▶ Riposiziona i file alla posizione *al momento del precedente commit* (cfr. esempio slide 60)

Commitment control for everyone

42



43

MK1 44
common
ITALY

Commit e rollback

Consolidare o annullare una transazione

Commitment control for everyone

44

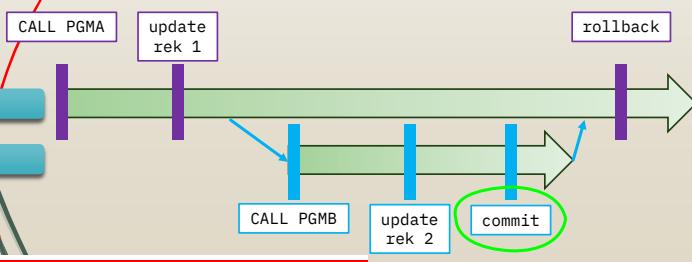
MK1
common
ITALY

45

Comm.lt. quiz

PGMA e PGMB vengono eseguiti entrambi nel *DFTACTGRP con COMMIT = *CHG

- PGMA esegue un update del record 1, quindi chiama il PGMB
- PGMB esegue un update del record 2, esegue commit e ritorna al chiamante
- PGMA esegue rollback



Quanti record vengono consolidati?

0 1 2



45

 MK1
common
ITALY

46

Esempio transazioni/1

PGMA -> PGMB
dft

- PGMA e PGMB vengono eseguiti entrambi nel *DFTACTGRP con COMMIT = *CHG
- PGMA esegue un update, quindi chiama il PGMB
- PGMB esegue un update di un altro record, esegue commit e ritorna al chiamante
- PGMA esegue rollback

Programma	Azione	Ambito ctl sincronia	ID LUW	Modifiche in sospeso
PGMA	update 1 record	*DFTACTGRP	APPN.sysname.X'48953C039E92'.00001	1 ✓
PGMB	update 1 record	*DFTACTGRP	APPN.sysname.X'48953C039E92'.00001	2 ✓ ↗
PGMB	commit	*DFTACTGRP	APPN.sysname.X'48953C039E92'.00002	0
PGMB	return			
PGMA	rollback	*DFTACTGRP	APPN.sysname.X'48953C039E92'.00003	0

RISULTATO: entrambi gli update vengono consolidati sul db dal commit eseguito da PGMB

Commitment control for everyone

TCMT1A.SQLRPLE, TCMT1B.SQLRPLE



46

47

Comm.lt. quiz

PGMA viene eseguito in *DFTACTGRP e PGMB in ACTGRP(*NEW) con COMMIT = *CHG

- PGMA esegue un update del record 1, quindi chiama il PGMB
- PGMB esegue un update del record 2, esegue commit e ritorna al chiamante
- PGMA esegue rollback

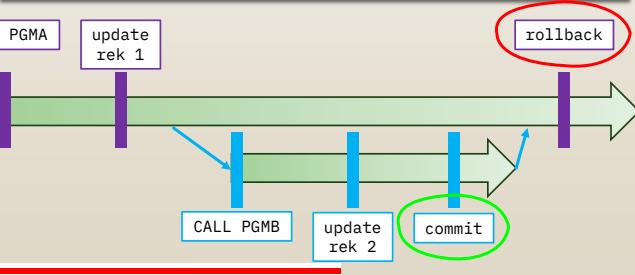


Quanti record vengono consolidati?

0 1 2



47



48

Esempio transazioni/2

PGMA
dft

PGMB
new

- PGMA viene eseguito in *DFTACTGRP e PGMB in ACTGRP(*NEW) con COMMIT = *CHG

PGMA esegue un update, quindi chiama il PGMB

PGMB esegue un update di un altro record, esegue commit e ritorna al chiamante

PGMA esegue rollback

Programma	Azione	Ambito ctl sincronia	ID LUW	Modifiche in sospeso
PGMA	update 1 record	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00001	1 ✗
PGMB	update 1 record	act group 19	APPN.sysname.X'5A75E299C288'.00001	1 ✓
PGMB	commit	act group 19	APPN.sysname.X'5A75E299C288'.00002	0
PGMB	return	<i>si chiude automaticamente act group 19 perché gestito dal sistema</i>		
PGMA	rollback	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00002	0

RISULTATO: solo l'update eseguito da PGMB viene consolidato sul db

TCMT2A.SQLRPGLE, TCMT2B.SQLRPGLE



48

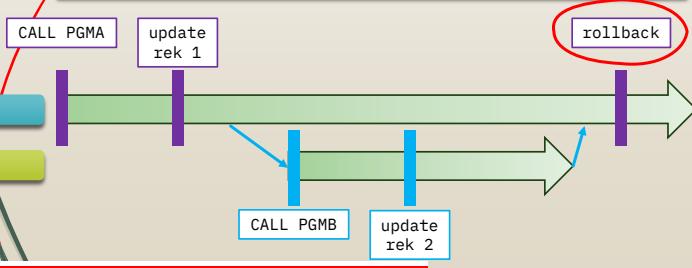
MK1
common
ITALY

49

Comm.lt. quiz

PGMA viene eseguito in *DFTACTGRP e PGMB in ACTGRP(*NEW) con COMMIT = *CHG

- PGMA esegue un update del record 1, quindi chiama il PGMB
- PGMB esegue un update del record 2 e ritorna al chiamante
- PGMA esegue rollback



Quanti record vengono consolidati?

0 1 2



Chi ha consolidato il record n. 2?

49

MK1
common
ITALY

50

Esempio transazioni/3

PGMA
dft

PGMB
new

- PGMA viene eseguito in *DFTACTGRP e PGMB in ACTGRP(*NEW) con COMMIT = *CHG

PGMA esegue un update, quindi chiama il PGMB

PGMB esegue un update di un altro record, NON esegue commit esplicito e ritorna al chiamante

PGMA esegue rollback

Programma	Azione	Ambito ctl sincronia	ID LUW	Modifiche in sospeso
PGMA	update 1 record	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00001	1 ✗ ↗
PGMB	update 1 record	act group 19	APPN.sysname.X'5A75E299C288'.00001	1 ✓ ↗
PGMB	return	si chiude act group 19 e viene eseguito un commit implicito		
PGMA	rollback	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00002	0 ↗

RISULTATO: solo l'update eseguito da PGMB viene consolidato sul db

Commitment control for everyone

* TCMT3A.SQLRPGLE, TCMT3B.SQLRPGLE



50

MK1
common
ITALY

51

Esempio transazioni/4

PGMA
dft

PGMB
name

- ▶ PGMA viene eseguito in *DFTACTGRP e PGMB in ACTGRP('PGMB') con COMMIT = *CHG
 - PGMA esegue un update, quindi chiama il PGMB
 - PGMB esegue un update di un altro record, esegue commit e ritorna al chiamante
 - PGMA esegue rollback

Programma	Azione	Ambito ctl sincronia	ID LUW	Modifiche in sospeso
PGMA	update 1 record	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00001	1 ✘
PGMB	update 1 record	act group PGMB	APPN.sysname.X'5A75E299C288'.00001	1 ✘
PGMB	return	<i>non si chiude act group PGMB perché non gestito dal sistema e la transazione rimane sospesa</i>		
PGMA	rollback	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00002	0

RISULTATO: l'update eseguito da PGMB rimane in sospeso fino a quando non viene chiuso l'actgrp PGMB

TCMT4A.SQLRPGLE, TCMT4B.SQLRPGLE

51

MK1
common
ITALY

52

Esempio transazioni/5

PGMA
dft

PGMB
new

- ▶ PGMA viene eseguito in *DFTACTGRP e PGMB in ACTGRP(*NEW) con COMMIT = *CHG
 - PGMA esegue un update, quindi chiama il PGMB con estensore errore CALLP (E)
 - PGMB esegue un update di un altro record e prima di eseguire commit si interrompe per un'eccezione non prevista, quindi ritorna a PGMA che prosegue all'istruzione successiva
 - PGMA esegue rollback

Programma	Azione	Ambito ctl sincronia	ID LUW	Modifiche in sospeso
PGMA	update 1 record	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00001	1 ✘
PGMB	update 1 record	act group 19	APPN.sysname.X'5A75E299C288'.00001	1 ✘
PGMB	eccezione 🐞	<i>si chiude act group 19 e viene eseguito un rollback implicito</i>		
PGMA	rollback	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00002	0

RISULTATO: nessun update viene consolidato sul db

TCMT5A.SQLRPGLE, TCMT5B.SQLRPGLE

52

53

Comm.lt. quiz

PGMA viene eseguito in *DFTACTGRP e PGMB in ACTGRP(*NEW) con COMMIT = *CHG

- PGMA esegue un update del record 1, quindi chiama il PGMB
- PGMB esegue un update del record 2 e ritorna al chiamante
- PGMA si chiude normalmente
- viene chiuso normalmente il job (signoff)



Quanti record vengono consolidati?

- 0 1 2
-

53

54

Esempio transazioni/6

PGMA
dft

PGMB
new

► PGMA viene eseguito in *DFTACTGRP e PGMB in ACTGRP(*NEW) con COMMIT = *CHG

PGMA esegue un update, quindi chiama il PGMB

PGMB esegue un update di un altro record, NON esegue commit esplicito e ritorna al chiamante

PGMA si chiude normalmente senza eseguire commit e poi viene chiuso normalmente il job

Programma	Azione	Ambito ctl sincronia	ID LUW	Modifiche in sospeso
PGMA	update 1 record	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00001	1 ✗
PGMB	update 1 record	act group 19	APPN.sysname.X'5A75E299C288'.00001	1 ✓ ↘
PGMB	return	<i>si chiude act group 19 e viene eseguito un commit隐式</i>		
PGMA	return	<i>transazione rimane sospesa con il rek aggiornato da PGMA ancora allocato</i>		
==	endjob	<i>si chiude il job e viene eseguito un rollback隐式</i>		

RISULTATO: solo l'update eseguito da PGMB viene consolidato sul db

TCMT6A.SQLRPGLE, TCMT6B.SQLRPGLE



54

MK1
common
ITALY

55

Esempio transazioni/7

PGMA -> PGMB
new - caller

- PGMA viene eseguito in ACTGRP(*NEW) e PGMB in ACTGRP(*CALLER) con COMMIT = *CHG

PGMA esegue un update, quindi chiama il PGMB

PGMB esegue un update di un altro record, NON esegue commit esplicito e ritorna al chiamante
PGMA esegue commit

Programma	Azione	Ambito ctl sincronia	ID LUW	Modifiche in sospeso
PGMA	update 1 record	act group 19	APPN.sysname.X'48953C039E92'.00001	1 ✓ ↘
PGMB	update 1 record	act group 19	APPN.sysname.X'48953C039E92'.00001	1 ✓ ↘
PGMB	return	<i>NON si chiude act group 19 e quindi non viene eseguita nessuna azione implicita</i>		
PGMA	commit	act group 19	APPN.sysname.X'48953C039E92'.00002	0
PGMA	return	<i>si chiude act group 19 e viene eseguito un commit implicito anche se non eseguito esplicitamente da pgmA</i>		

RISULTATO: entrambi gli update vengono consolidati sul db dal commit eseguito da PGMA

* TCMT7A.SQLRPGLE, TCMT7B.SQLRPGLE

55

MK1
common
ITALY

56

Comm.lt. quiz

PGMA viene eseguito in ACTGRP(*NEW) e PGMB in *DFTACTGRP con COMMIT = *CHG

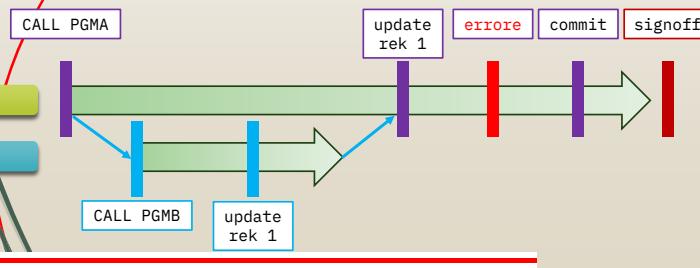
- PGMA chiama il PGMB
- PGMB esegue update del record 1 e ritorna al chiamante
- PGMA tenta di eseguire update sullo stesso record modificato da PGMB



Quanti record vengono consolidati?

0
↑

1



56

57

Esempio transazioni/8

PGMA
new

PGMB
dft

- PGMA viene eseguito in ACTGRP(*NEW) e PGMB in *DFTACTGRP con COMMIT = *CHG
- PGMA chiama il PGMB; PGMB esegue un update di un record ma NON esegue commit esplicito e ritorna al chiamante; PGMA tenta di eseguire update sullo stesso record modificato da PGMB

Programma	Azione	Ambito ctl sincronia	ID LUW	Modifiche in sospeso
PGMA	si apre act group 22		APPN.sysname.X'5A779697C288'.00001	
PGMB	update record n	*DFTACTGRP	APPN.sysname.X'5A77D18EC288'.00001	1 ✗
PGMB	return		NON si chiude dftactgrp e non viene eseguita nessuna azione implicita transazione rimane sospesa con il rek aggiornato da PGMB allocato	
PGMA	update record n	act group 22	errore record allocato	🐞
PGMA	commit	act group 22	APPN.sysname.X'5A779697C288'.00002	
PGMA	return		si chiude act group 22 ma la transazione del *dftactgrp rimane sospesa con il rek aggiornato da PGMB ancora allocato	
==	endjob		si chiude il job e viene eseguito un rollback implicito	

RISULTATO: l'update di PGMA fallisce, l'update di PGMB rimane in sospeso fino a che si chiude il job

TCMT8A.SQLRPGLE, TCMT8B.SQLRPGLE



57

58

Esempio transazioni/9

PGMA
new

PGMB
new

- PGMA e PGMB vengono eseguiti in due ACTGRP(*NEW) con COMMIT = *CHG
- PGMA esegue un update, quindi chiama il PGMB
- PGMB esegue un update di un altro record, esegue commit e ritorna al chiamante
- PGMA esegue commit

Programma	Azione	Ambito ctl sincronia	ID LUW	Modifiche in sospeso
PGMA	update 1 record	act group 21	APPN.sysname.X'5A6EF3A9C263'.00001	1 ✓ ↘
PGMB	update 1 record	act group 22	APPN.sysname.X'5A6FE4BBC263'.00001	1 ✓ ↘
PGMB	commit	act group 22	APPN.sysname.X'5A6FE4BBC263'.00002	0 ↗
PGMA	commit	act group 21	APPN.sysname.X'5A6EF3A9C263'.00002	0 ↗

RISULTATO: ogni update eseguito da PGMA e PGMB viene consolidato sul db dal proprio commit in modo indipendente

Commitment control for everyone

TCMT9A.SQLRPGLE, TCMT9B.SQLRPGLE

58

Commit o rollback implicito

Contesto	Azione implicita
Programma si interrompe con eccezione non prevista	rollback
Gruppo di attivazione si chiude normalmente	commit
Gruppo di attivazione si chiude con errori	rollback
Definizioni di commit in ambito *JOB o *DFTACTGRP	nessuna
Chiusura di un lavoro *JOB normale o anomala	rollback

<https://www.ibm.com/docs/en/i/7.6.0?topic=control-system-initiated-end-commitment>

Commitment control for everyone

To be continued...



Commitment control for everyone