







Marco Riva

www.markonetools.it

IBM CHAMPION

2021-2025

Privacy area









Ultimo aggiornamento: 01/02/2026

1







 <div>Critical</div>	 <div>Important</div>	 <div>Nice to have</div>
<div>Gruppi attivazione</div> <div>Ambito</div> <div>Attivazione</div> <div>Operazioni implicite</div>	<div>Livello isolamento</div> <div>Concorrenza accesso</div> <div>Monitoraggio</div>	<div>Deadlock</div> <div>Savepoint</div> <div>Soft commit</div> <div>Giornali</div>

Commitment control for everyone

2

MK1

common

ITALY

3

21/gen

1

4/feb

2

18/feb

3

4/mar

4

Sommario

Un ripasso di job e activation group

Concetti base del controllo di sincronia e glossario

Prerequisiti

Attivare il controllo di sincronia

Ambito del controllo di sincronia

Perché usare il controllo di sincronia

Livello di isolamento

Commit e rollback impliciti

Esempi di transazioni

Concorrenza di accesso

Cursori SQL vs commit/rollback

Savepoint

Monitoraggio del controllo sincronia

Controllo di sincronia e voci di giornale

Istruzioni SQL per i giornali

Commitment control for everyone

MK1

common

ITALY

4

Esempi e documentazione del corso

<https://github.com/mk1tools/Commitment-control-tutorial.git>



GitHub

MK1

Commitment control for everyone

Controllo di sincronia

Concetti base

Commitment control for everyone

5

Controllo sincronia (commitment control)

COSA

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

COME

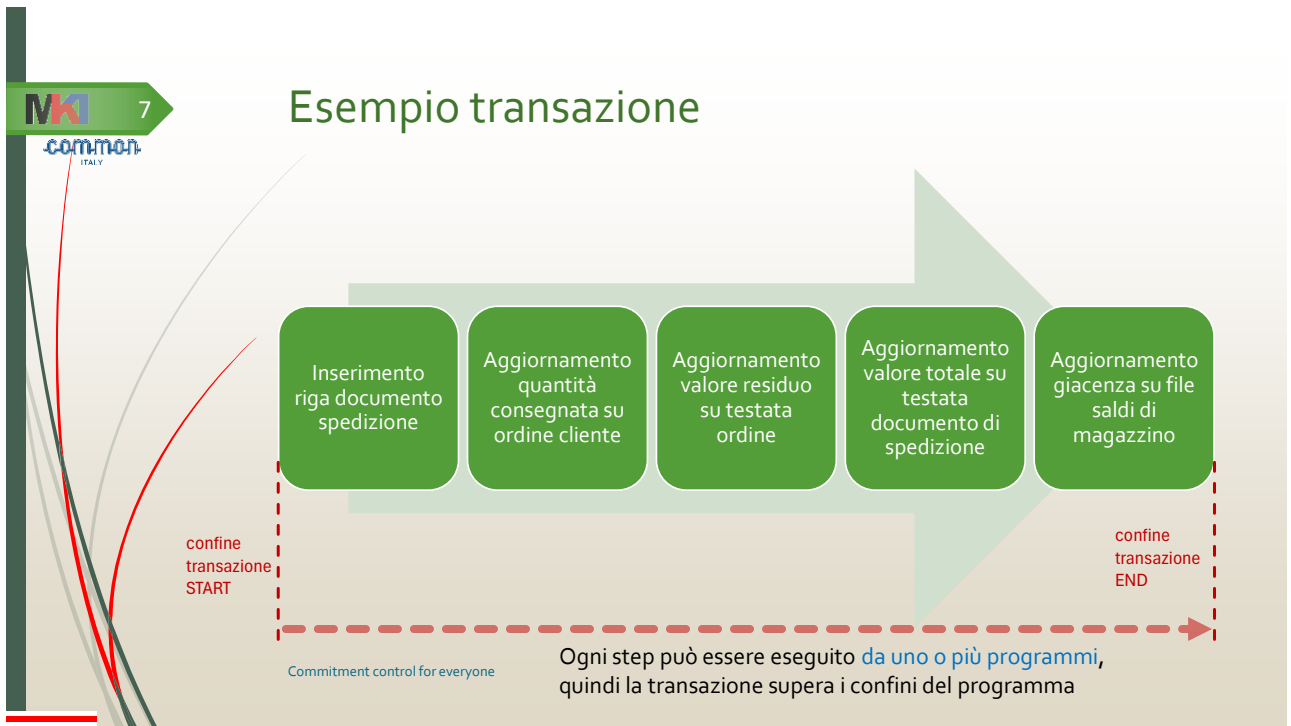
- **transazione**
- è un gruppo di modifiche a uno o più file di database che dal punto di vista dell'utente appaiono come una singola modifica

PERCHÉ

- **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

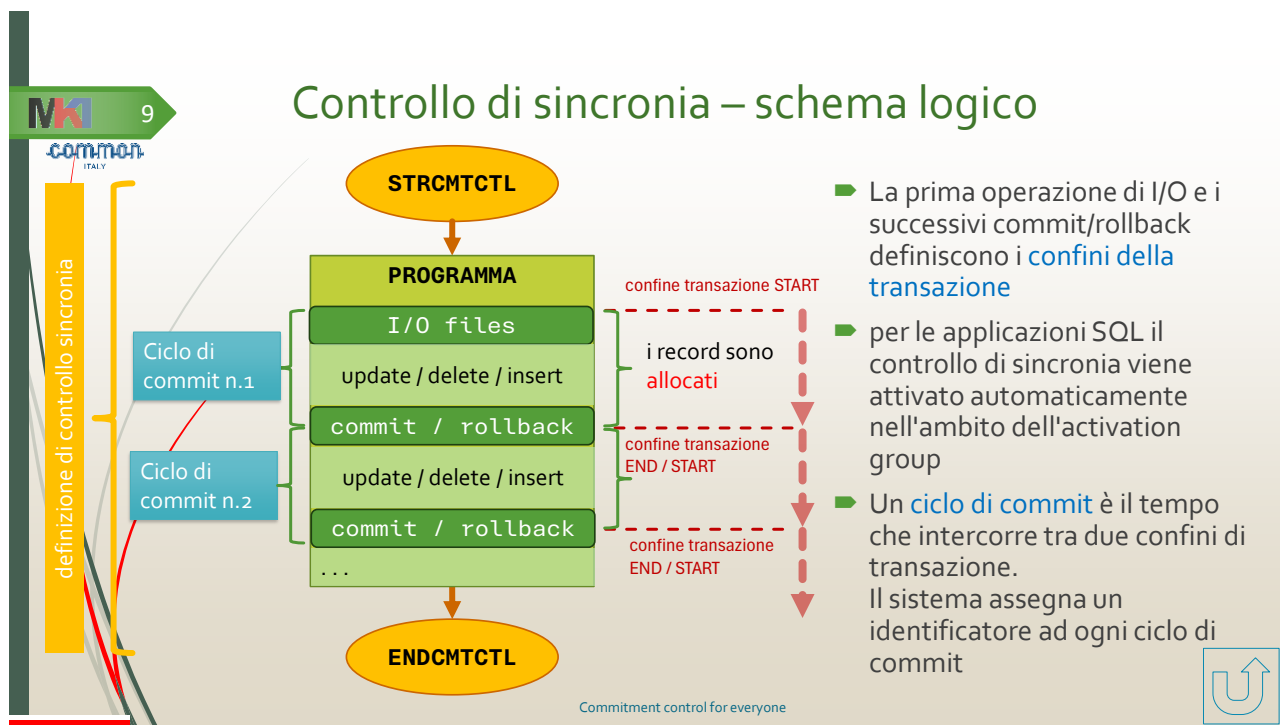
6



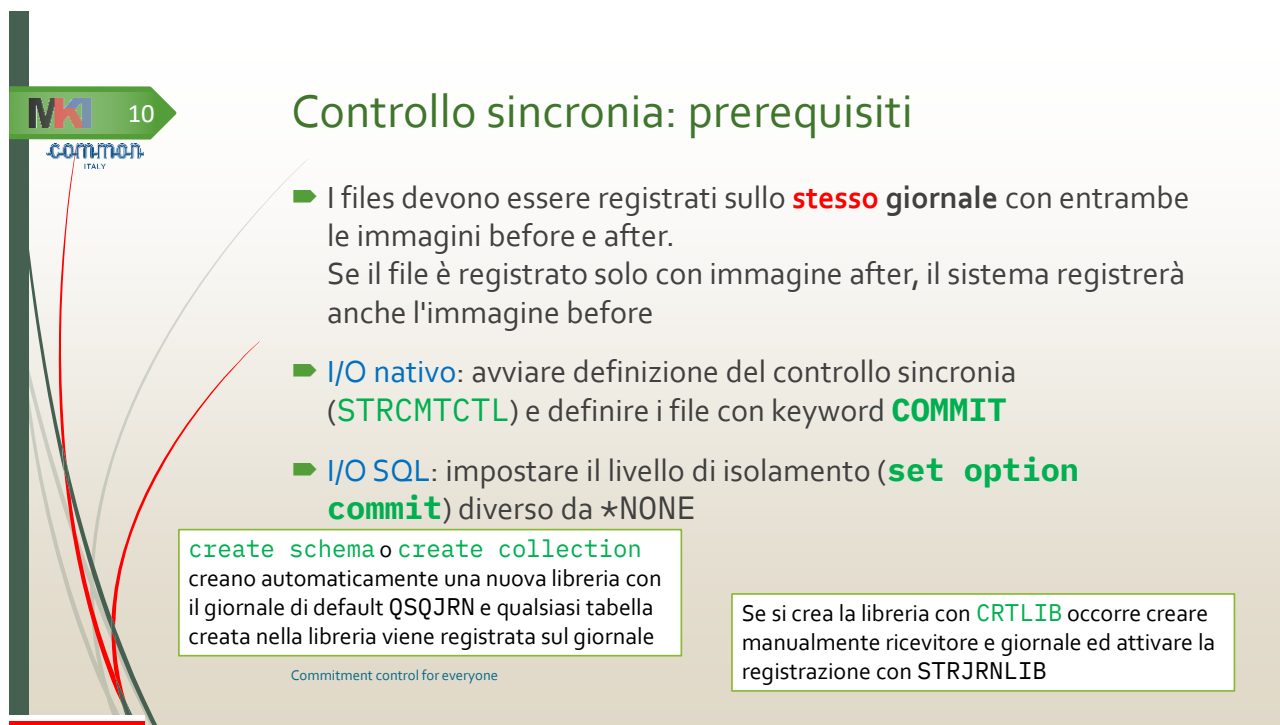
7



8



9



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

MK 13

common
ITALY

Job e activation group

Un breve ripasso

Commitment control for everyone

13

MK 14

common
ITALY

Job e activation group: program isolation

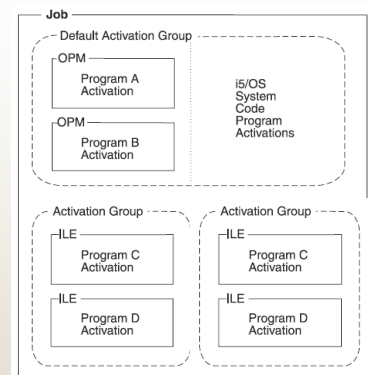
- L'**activation group** contiene le **risorse** necessarie ad eseguire il programma.

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 compatibile



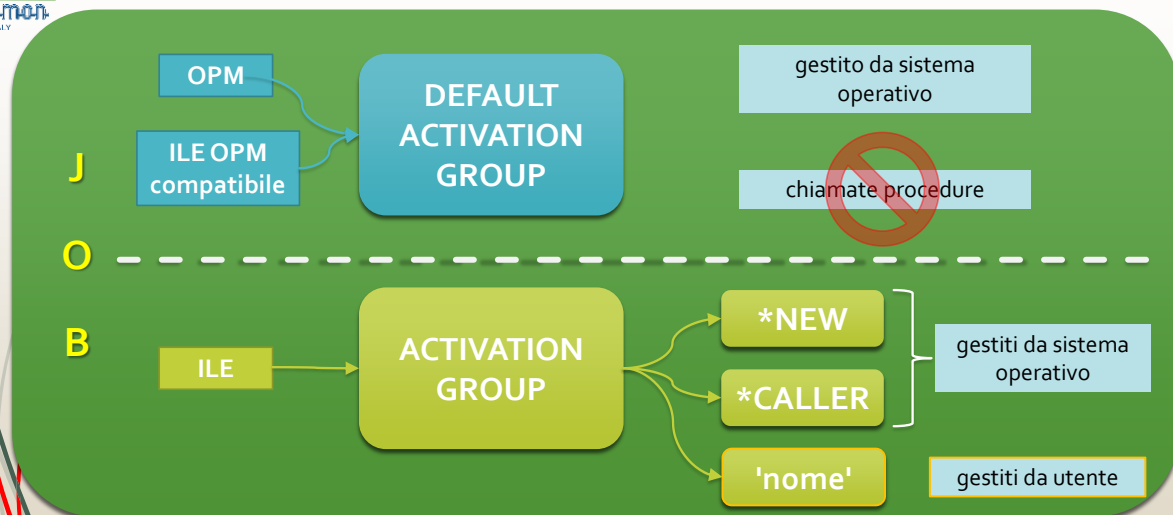
Fonte IBM

Commitment control for everyone

14

MK 15

Activation group in sintesi



Commitment control for everyone

15

MK 16

Activation group: comandi principali

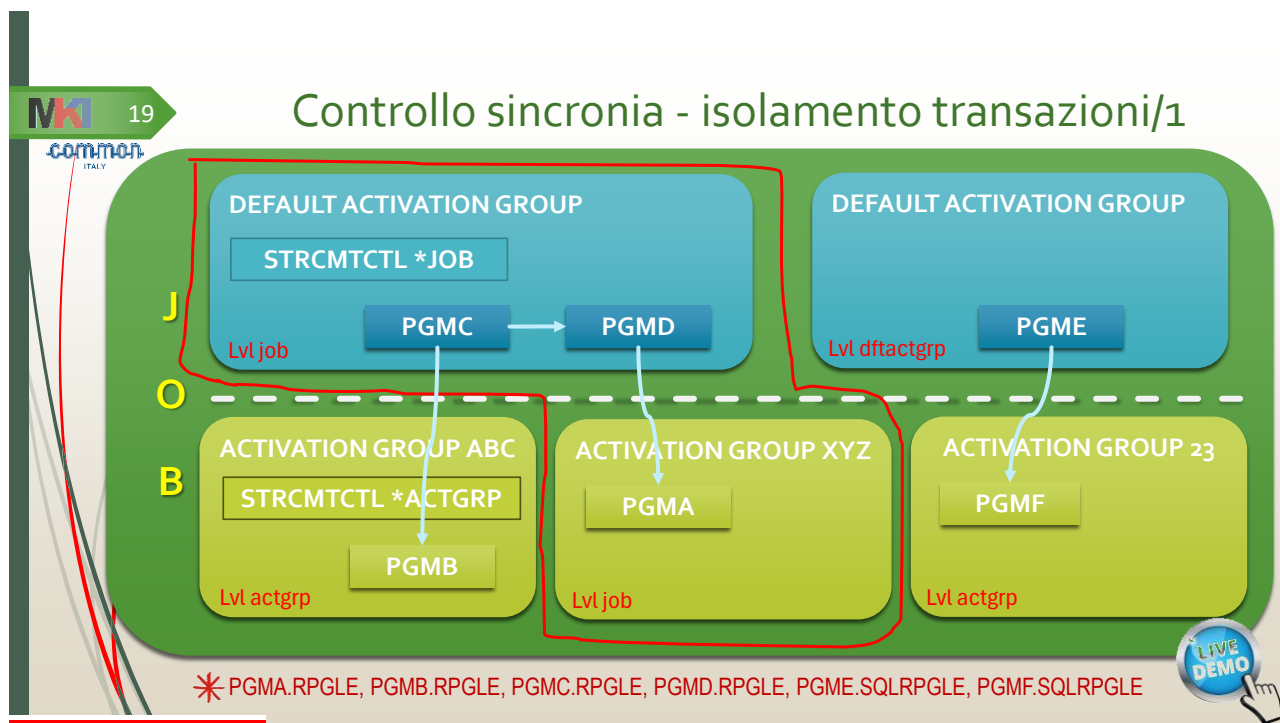
- **RCLACTGRP**: libera le risorse di un AG nominale
- **CRTSQLRPGI** parametro **CLOSQLCSR**: determina quando viene chiuso automaticamente un cursore SQL. *ENDACTGRP o *ENDMOD.
- **OVR...** parametro **OVRSCOPE**: determina l'ambito di validità della sostituzione. *ACTGRPDFN o *CALLLVL o *JOB.
- **DLTOVR**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

Commitment control for everyone

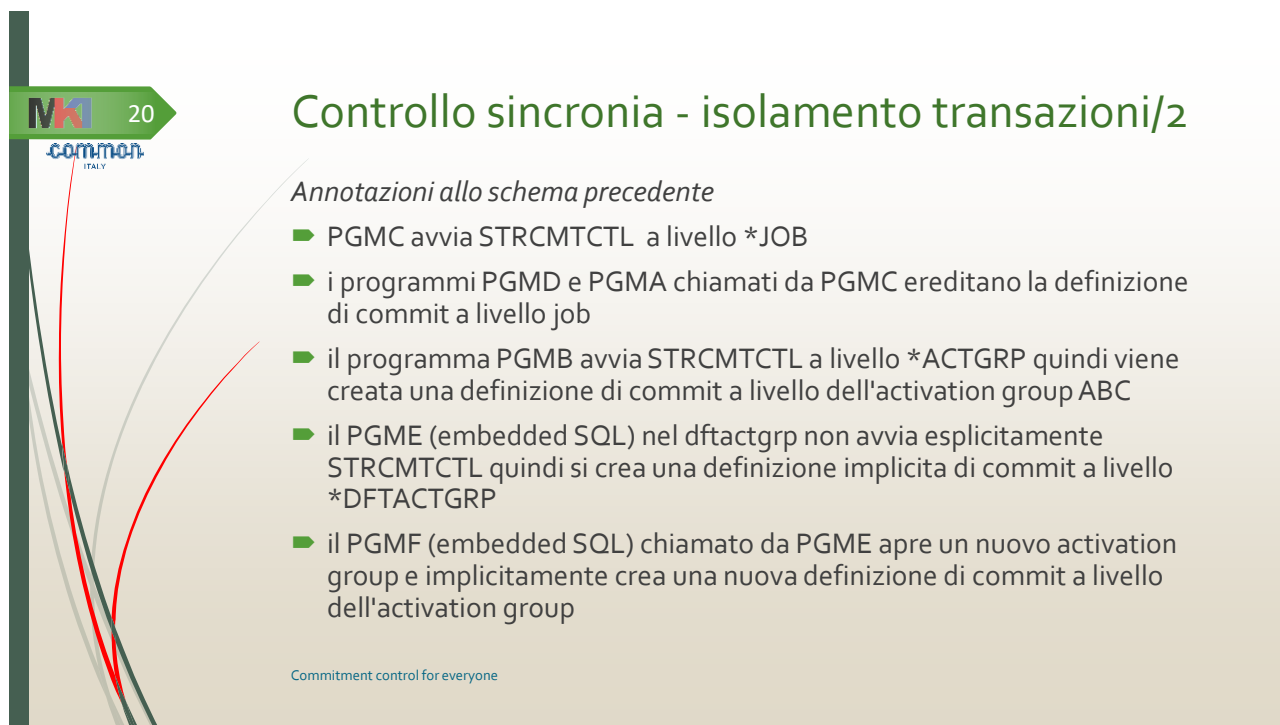
16

17

18



19



20

21

22

MK1

23

common

ITALY

PGMC

DSPLY

Avvio programma PGMD. Invio per proseguire.

Demo - CALL PGMC/3

WRKCMTDFN

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

Visualizzazione stack di chiamata

Lavoro: MRIVAA Utente: MRIVA Numero: 0000016A
Sottoproc.: 0000016A

-----Gruppo attivazione-----

Tipo	Programma	Nome	Numero
	QCMD QSYS	*DFTACTGRP	0000000000000001
	QUICMENU QSYS	*DFTACTGRP	0000000000000001
1	QUIMNDRV QSYS	*DFTACTGRP	0000000000000001
2	QUIMGFLW QSYS	*DFTACTGRP	0000000000000001
3	QUICMD QSYS	*DFTACTGRP	0000000000000001
	PGMC MK1SQL	*DFTACTGRP	0000000000000002
	PGMD MK1SQL	*DFTACTGRP	0000000000000002
	PGMD MK1SQL	*DFTACTGRP	0000000000000002

Commitment control for everyone

23

MK1

24

common

ITALY

PGMC

DSPLY

Eseguito update 000210. Proseguire? (S/N)

Demo - CALL PGMC/4

WRKCMTDFN

Visualizz. stato livello record

Lavoro: MRIVAA Utente: MRIVA Numero: 460667 Sistema: 460667
Definizione commit : *JOB

-----Modifiche-----

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

Id ciclo commit

File	Libreria	Membro	Id ciclo commit
EMPLOYEE	MK1SAMPLE	EMPLOYEE	6727
EMPLOYEE	EMPLOYEE	EMPLOYEE	6727

Opz Giornale Libr. Id ciclo commit Blocchi di record Modifiche in sospeso

QSQRN MK1SAMPLE 6727 2 2

Commitment control for everyone

24

MarkOneTools - MK1

MK1

25

common

ITALY

PGMA

DSPLY Avvio programma PGMA. Invio per proseguire.

Visualizzazione stack di chiamata

Lavoro: MRIVA Utente: MRIVA Numero: 0000016A

Sottoproc.: 0000016A

-----Gruppo attivazione-----

Tipo	Programma	Nome	Numero
	QCMD	QSYS	*DFTACTGRP 0000000000000001
	QUICMENU	QSYS	*DFTACTGRP 0000000000000001
1	QUIMNDRV	QSYS	*DFTACTGRP 0000000000000001
2	QUINGFLW	QSYS	*DFTACTGRP 0000000000000001
3	QUICMD	QSYS	*DFTACTGRP 0000000000000001
	PGMC	MKISQL	*DFTACTGRP 0000000000000002
	PGMC	MKISQL	*DFTACTGRP 0000000000000002
	PGMD	MKISQL	*DFTACTGRP 0000000000000002
	PGMD	MKISQL	*DFTACTGRP 0000000000000002
	PGMA	MKISQL	*NEW 0000000000000013
	PGMA	MKISQL	*NEW 0000000000000013

Visualizzazione gruppo attivazione

Lavoro: MRIVA Utente: MRIVA Numero: 460667

Sistema: 460667

-----Gruppo attivazione-----

Nome	Numero	Indicatore in uso	Programma	Libreria
*DFTACTGRP	0000000000000001	SI		
*DFTACTGRP	0000000000000002	SI		
QLGLOCAL	0000000000000011	Senza	QLGLOCAL	QSYS
QSQCCLI	0000000000000012	Senza	QSQCCLI	QSYS
19	0000000000000013	SI	PGMA	MKISQL

-----Modifiche-----

File	Libreria	Membro	Commit	Rollback	Sospeso
EMPLOYEE	MKISAMPLE	EMPLOYEE	0	0	1
EMPLOYEE	EMPLOYEE	EMPLOYEE	0	0	1
EMPLOYEE	EMPLOYEE	EMPLOYEE	0	0	1

File Libreria Membro Id ciclo commit

EMPLOYEE MKISAMPLE EMPLOYEE 6727

EMPLOYEE EMPLOYEE 6727

Commitment control for everyone

25

MK1

26

common

ITALY

PGMB

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

Visualizzazione stack di chiamata

Lavoro: MRIVA Utente: MRIVA Numero: 0000016A

Sottoproc.: 0000016A

-----Gruppo attivazione-----

Tipo	Programma	Nome	Numero
	QCMD	QSYS	*DFTACTGRP 0000000000000001
	QUICMENU	QSYS	*DFTACTGRP 0000000000000001
1	QUIMNDRV	QSYS	*DFTACTGRP 0000000000000001
2	QUINGFLW	QSYS	*DFTACTGRP 0000000000000001
3	QUICMD	QSYS	*DFTACTGRP 0000000000000001
	PGMC	MKISQL	*DFTACTGRP 0000000000000002
	PGMC	MKISQL	*DFTACTGRP 0000000000000002
	PGMB	MKISQL	*NEW 0000000000000014
	PGMB	MKISQL	*NEW 0000000000000014

Visualizzazione gruppo attivazione

Lavoro: MRIVA Utente: MRIVA Numero: 460667

Sistema: 460667

-----Gruppo attivazione-----

Nome	Numero	Indicatore in uso	Programma	Libreria
*DFTACTGRP	0000000000000001	SI		
*DFTACTGRP	0000000000000002	SI		
QLGLOCAL	0000000000000011	Senza	QLGLOCAL	QSYS
QSQCCLI	0000000000000012	Senza	QSQCCLI	QSYS
20	0000000000000014	SI	PGMB	MKISQL

Visualizzazione stato definizione commit

Lavoro: MRIVA Utente: MRIVA Numero: 460667

20/01/26 15:22:52

Sottoproc.: 0000016A

Sistema: 460667

-----Stato definizione commit-----

Sottoproc.	0000016A	*NONE
ID LUM (Logical Unit of Work)	APPN.BD21CSAF077B.00001	
ID spazio blocco	UDB_01000000000409F5	
Definizione commit	20	
Gruppo attivazione	20	
Gruppo ASP	*SYSBAS	
Ubicazione risorsa	NESSUNO	
Livello blocco predefinito	*CHG	
Utente	MRIVA	
Modifiche in sospeso locali	NO	
Ruolo	RIPRISTINO	
Stato	Registrazione data/ora	

Risincronizz. in corso

Operaz. euristica

Lavoro attivo

Lavoro server

Ambito blocco

Supero tempo transazione

Attesa massima blocco

NO

*CHG

SI

*ACTGRP

26

27

common

ITALY

PGMB

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

Demo - CALL PGMC/7

WRKCMTDFN

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

Visualizz. stato livello record

Lavoro: MK1SAMPLE Utente: MK1SAMPLE Sistema: 460667

Definizione commit : 20

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

Opz	Giornale	Libr.	Id ciclo commit	Blocchi di record	Modifiche in sospeso
QSQJRN		MK1SAMPLE	6735	1	1

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	MK1SAMPLE	MK1SAMPLE	15:25:09
6736	R	UB	EMPLOYEE	MK1SAMPLE	MK1SAMPLE	15:25:09
6737	R	UP	EMPLOYEE	MK1SAMPLE	MK1SAMPLE	15:25:09

Commitment control for everyone

27

28

common

ITALY

PGMB

DSPLY PGMB ha eseguito commit. Invio per proseguire

Demo - CALL PGMC/8

WRKCMTDFN

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

-----Modifiche-----

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

Visualizzazione stato definizione commit

20/01/26 15:31:25

Lavoro: MK1SAMPLE Utente: MK1SAMPLE Numero: 460667

Sottopr. : *NONE

ID LUW (Logical Unit of Work) : APPN.X'BD21C5AF077B'.00002

ID spazio blocco : UDB_0100000000409F5

Definizione commit : 20

Gruppo attivazione : 20

Gruppo ASP : *SYSBAS

Limite blocco : 50000000

Commit durevole : SI

Numero di commit : 1

Numero di rollback : 0

Giornale predefinito : 0

Libreria :

Commitment control for everyone

28

MarkOneTools - MK1

29

common

PGMB

DSPLY PGMB arresterà il contr.sincr. Invio per proseguire

DSPLY Arresto controllo di sincronia con errori

ENDCMCTL 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

Demo - CALL PGMC/9

WRKCMTDFN

Visualizz. stato livello record

Lavoro: MRIVAA Utente: MRIVA Sistema: AS1PRD Numero: 460667

Definizione commit : 20

File	Libreria	Membro	Liv blocco	Stato	Risoluzione accessi contemp.
EMPLOYEE	MK1SAMPLE	EMPLOYEE	*CHG	APERTO	*WAIT

Visualizz. stato livello record

Lavoro: MRIVAA Utente: MRIVA Sistema: AS1PRD Numero: 460667

Definizione commit : 20

File	Libreria	Membro	Liv blocco	Stato	Risoluzione accessi contemp.

(Non ci sono modifiche a livello record sotto controllo di commit.)

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

Commitment control for everyone

29

30

common

PGMC

DSPLY PGMC arresterà il contr.sincr. Invio per proseguire

DSPLY Arresto controllo di sincronia con errori

Demo - CALL PGMC/10

WRKCMTDFN

Visualizz. stato livello record

Lavoro: MRIVAA Utente: MRIVA Sistema: AS1PRD Numero: 460667

Definizione commit : *JOB

File	Libreria	Membro	Liv blocco	Stato	Risoluzione accessi contemp.
EMPLOYEE	MK1SAMPLE	EMPLOYEE	*CHG	APERTO	*WAIT
				CHIUSO	

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

Definiz. Opz	commit	N° lav.	Uten. lav.	Nome lav.	Sottopr.	Utente

(Non ci sono definizioni di commit attive)

Chiusura PGMC

SIGNOFF

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.

5779551 V7R5M0 220415 Registrazione lavoro

Nome lavoro : MRIVAA Utente : MRIVA AS1PRD 20/01/26 15:41:04 CET Pag. 14

Descrizione lavoro : QDFTJOB0 Libreria : QGPL Numero : 460667

IDMSG TIPO GRAV DATA ORA DA PGM LIBRERIA 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 LUM (logical unit of work/unità logica di lavoro) APPN.AS1PRD.X'BCFEBB5E077B'.00001.

30

MarkOneTools - MK1

31

common

ITALY

Comm.It. quiz

J
O
B

DEFAULT ACTIVATION GROUP

STRCMTCTL *JOB

PGMG

STRCMTCTL *ACTGRP

PGMH

* PGMG.RPGLE, PGMH.RPGLE

QUIZ TIME?

3

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

YES

NO

↑

31

32

common

ITALY

Controllo sincronia - isolamento transazioni/3

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

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

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

Commitment control for everyone

32

33

common
ITALY

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

34

common
ITALY

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. Alloca 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 consolidate.
*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.o?topic=concepts-isolation-level>

34

MK1

35

common

ITALY

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.o?topic=level-comparison-isolation-levels>

Commitment control for everyone

35

MK1

36

common

ITALY

Livello di allocazione

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

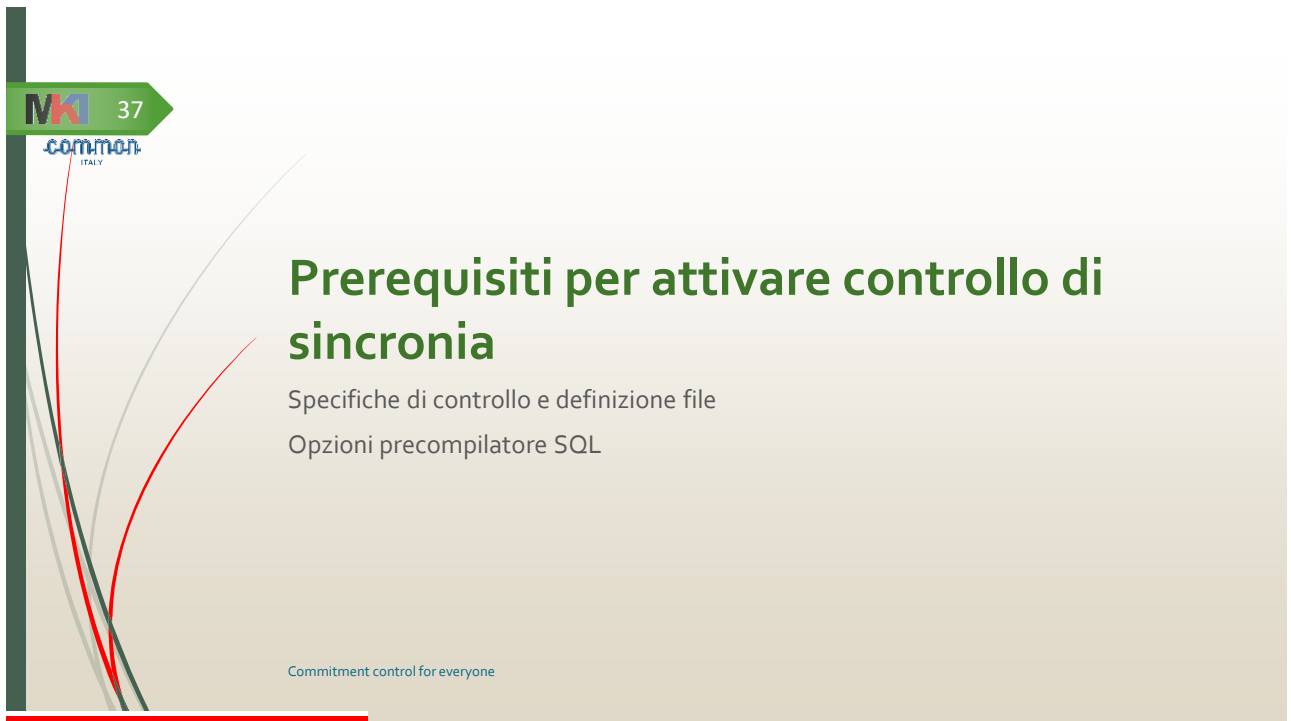
Tipo	Descrizione	Conseguenze
*ALL	allocazione dei record modificati e letti durante tutta la transazione	- anche un record letto senza scopo di aggiornamento è allocato
*CS	allocazione dei record modificati durante tutta la transazione allocazione dei record letti fino al rilascio o alla successiva lettura	- 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	

<https://www.ibm.com/docs/en/i/7.6.o?topic=control-commit-lock-level>

Commitment control for everyone

integrità e sicurezza del dato

36

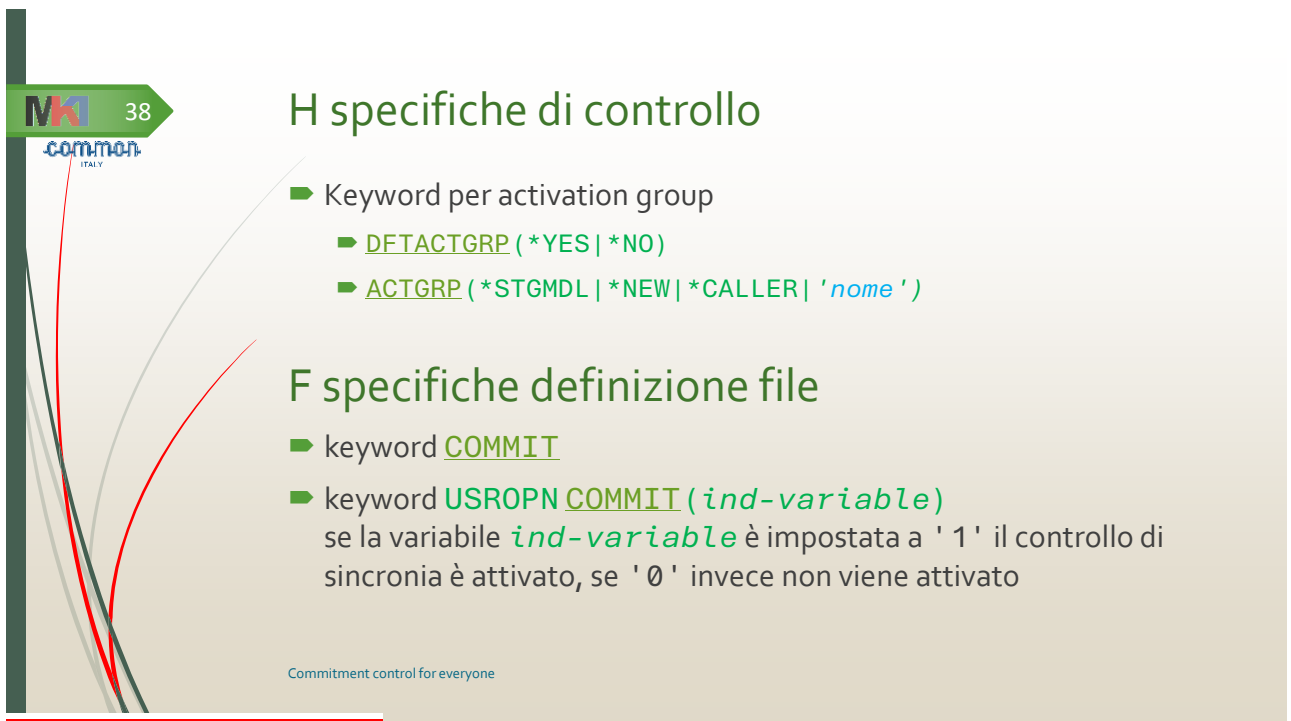


Prerequisiti per attivare controllo di sincronia

- 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



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



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.o?topic=statement-isolation-clause>

40

MK 41

common
ITALY

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

MK 42

common
ITALY

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

 44
common
ITALY

Commit e rollback

Consolidare o annullare una transazione

Commitment control for everyone

44

45

common

Comm.It. 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

CALL PGMA

update rek 1

CALL PGMB

update rek 2

commit

rollback

4

QUIZ TIME?

Quanti record vengono consolidati?

0

1

2

↑

45

46

common

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.SQLRPGLE, TCMT1B.SQLRPGLE

LIVE DEMO

46

MarkOneTools - MK1

47

common

Comm.It. 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

CALL PGMA

update rek 1

CALL PGMB

update rek 2

commit

rollback

QUIZ TIME?

5

Quanti record vengono consolidati?

0

1

2

↑

47

48

common

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	si chiude automaticamente act group 19 perché gestito dal sistema			
PGMA	rollback	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00002	0	

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

* TCMT2A.SQLRPGLE, TCMT2B.SQLRPGLE

LIVE DEMO

48

49

common

Comm.It. 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

CALL PGMA

update rek 1

CALL PGMB

update rek 2

rollback

QUIZ TIME?

6

Quanti record vengono consolidati?

0

1

2

↑

Chi ha consolidato il record n. 2?

49

50

common

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

LIVE DEMO

50

MarkOneTools - MK1

51

common

PGMA

dft

PGMB

name

Esempio transazioni/4

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	non si chiude act group PGMB perché non gestito dal sistema e la transazione rimane sospesa		
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

52

common

PGMA

dft

PGMB

new

Esempio transazioni/5

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 🐛	si chiude act group 19 e viene eseguito un rollback implicito		
PGMA	rollback	*DFTACTGRP	APPN.sysname.X'5A75B9A8C288'.00002	0

RISULTATO: nessun update viene consolidato sul db

✖ TCMT5A.SQLRPGLE, TCMT5B.SQLRPGLE

52

53

common

Comm.It. 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)

CALL PGMA

update rek 1

CALL PGMB

update rek 2

signoff

QUIZ TIME?

7

Quanti record vengono consolidati?

0

1

2

↑

53

54

common

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	si chiude act group 19 e viene eseguito un commit implicito		
PGMA	return	transazione rimane sospesa con il rek aggiornato da PGMA ancora allocato		
==	endjob	si chiude il job e viene eseguito un rollback implicito		

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

* TCMT6A.SQLRPGLE, TCMT6B.SQLRPGLE

LIVE DEMO

54

55

common

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 sospenso
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	NON si chiude act group 19 e quindi non viene eseguita nessuna azione implicita		
PGMA	commit	act group 19	APPN.sysname.X'48953C039E92'.00002	0
PGMA	return	si chiude act group 19 e viene eseguito un commit implicito anche se non eseguito esplicitamente da pgma		

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

* TCMT7A.SQLRPGLE, TCMT7B.SQLRPGLE

55

56

common

Comm.It. 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

CALL PGMA

update rek 1

errore

commit

signoff

CALL PGMB

update rek 1

Quanti record vengono consolidati?

0

1

56

MarkOneTools - MK1

57

common

ITALY

PGMA

new

PGMB

dft

Esempio transazioni/8

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

LIVE DEMO

57

58

common

ITALY

PGMA

new

PGMB

new

Esempio transazioni/9

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

59

common

ITALY

Commitment control for everyone

✖

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.o?topic=control-system-initiated-end-commitment>

Commitment control for everyone

59

60

common

ITALY

Commitment control for everyone

60