Föreläsning 5 – Säkerhet och transaktioner

725G28: Databaser och datamodellering

Jonathan Crusoe
Informatik, IEI, Linköpings Universitet

LINKÖPINGS UNIVERSITET

Agenda

- Denna föreläsning behandlar aspekter som kan uppkomma efter det att databasen designats och implementerats.

- Säkerhet & Behörighet 

- Transaktioner & Återställning 

Padron-McCarthy och Risch (2005), Databasteknik, Kapitel 14, 24 och 25.

music					
music_num	**song_title**	**artist**	**album**	**review_rank**	**price**
1	Trump Bing Bong Remixes	Bosco	YouTube	8 of 10	10$
2	I'm an Albatraoz	AronChupa	I'm an Albatraoz	1 of 5	 $
3	Fanten	Jonatan Ersarp	Unknown	3 of 7	1$
…	…	…	…	…	…

movies			
movie_num	**title**	**review_rank**	**price**
1	I det hetaste laget	5 of 5	10$
2	Frozen	3 of 5	5$
3	Star Trek: Discovery	2 of 6	1$
…	…	…	…

sales				
sales_num	**sales_date**	**discount**	**staff_num**	**customer_num**
1	Maj-17	10	1	2
2	Jun-20	0	4	5
3	Apr-12	100	1	18
…	…	…	…	

books			
book_num	**title**	**review_rank**	**price**
1	On Rhetoric	? of 5	323 kr
2	A Rulebook for Arguments	4 of 5	114 kr
3	Quran	4 of 5	54 kr
4	Rules For Radicals	5 of 5	149 kr
5	The Torah	? of 5	227 kr
6	The Prince	3 of 5	123 kr
…	…	…	…

sale_items				
sales_num	**music_num**	**movie_num**	**book_num**	**quantity**
1	1	null		2
1	null	null	2	10
2	null	3		1
…	…	…		…

people					
people_num	**email**	**first_name**	**last_name**	**gender**	**birth_date**
1	rock@gmail.	Frisky	McRock	M	52”
2	alter.pine@ya.	Donald	McGregor	F	00
3	fish@gmail.	Sotis	Crusoe	Fish	Fish
…	…	…	…	…	…

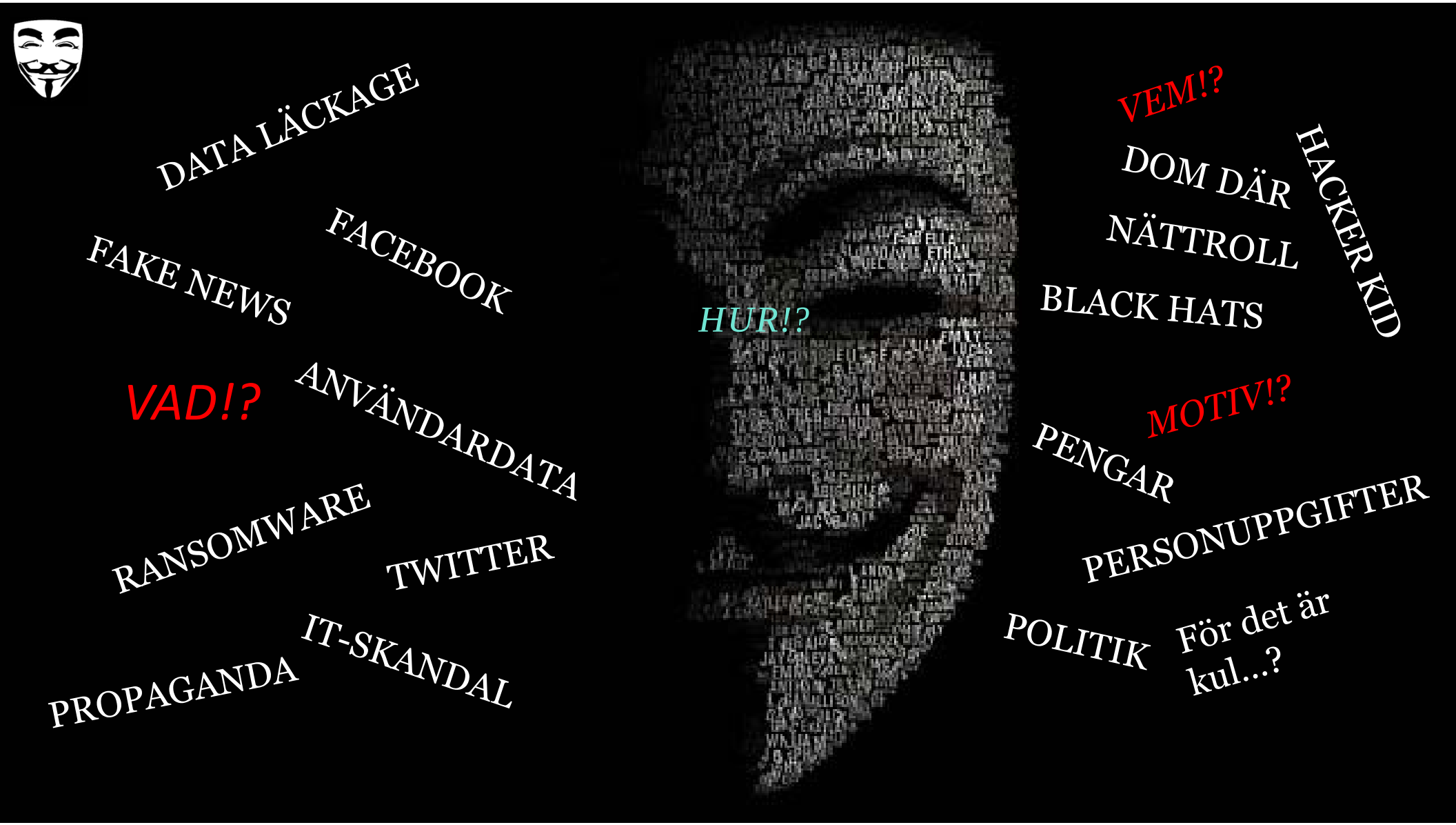
lives_at		
people_num	**add_num**	**date**
1	2	1990-05-21
2	3	1985-06-30
3	1	2000-01-22
…	…	

people_addresses					
add_num	**street**	**city**	**zip_code**	**floor**	**gate_code**
1	Dockyard 1	Stockholm	F124	0	F124
2	Desertroad 2	Cancas	506 43	6	2345
3	Trimergate 18	London	405 18	2	-
…	…	…	…	…	…

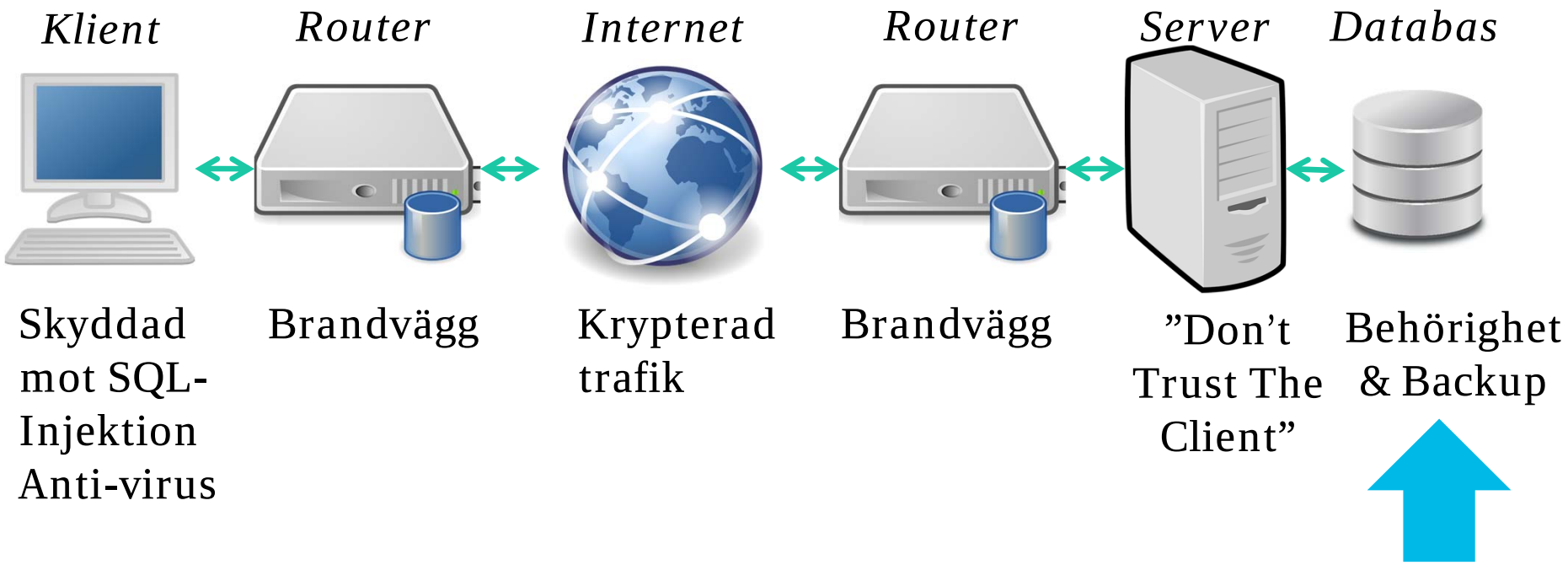
←
sales(staff_num, customer_num) pekar till denna tabell ☺

Säkerhet & Behörighet

LINKÖPINGS UNIVERSITET


DATA LÄCKAGE
FACEBOOK
FAKE NEWS
VAD!?
ANVÄNDARDATA
RANSOMWARE
TWITTER
IT-SKANDAL
PROPAGANDA
HUR!?
VEM!?
DOM DÄR
NÄTTROLL
BLACK HATS
HACKER KID
MOTIV!?
PENGAR
PERSONUPPGIFTER
POLITIK
För det är kul...?

Grov skiss över teknisksäkerhetsinfrastruktur



Lite Standard Säkerhet

Ingen säkerhet & alla har behörighet till allt…

Vi har data om…
Musik
Filmer
Böcker
Försäljningar
Personer
Adresser

- Varför är det dåligt?
- Vad kan utomstående göra med datan?

Behörighet (Användare)

- Användare med inloggning
  - Tabell, rad, och kolumn åtkomst
  - DML & DDL åtkomst
    - `SELECT`, `UPDATE`, `DELETE`, & `INSERT` + Andra (`DROP`, etc)

```
CREATE USER [username] WITH PASSWORD = '********';
DROP USER IF EXISTS [username];
ALTER USER [username] WITH ...;
```

Exempel för CREATE/ALTER/DROP USER

```
CREATE USER Sotis WITH PASSWORD = 'Fisk';
```
Skapar användaren Sotis

```
ALTER USER Sotis WITH NAME = FakeCat;
```
Byter namn till FakeCat

```
DROP USER IF EXISTS FakeCat;
```
Tar Bort FakeCat

Behörighet (Roller)

- Som användare fast lättare att arbeta med och underhålla
- Tilldelar användare roller
- Tilldelar rollerna vad de kan göra

```
CREATE ROLE [role_name];
ALTER ROLE [role_name] ADD/DROP MEMBER [role_name OR username];
DROP ROLE IF EXISTS [role_name];
```

Exempel för CREATE/ALTER/DROP ROLE

```
CREATE ROLE Customer;
```
Skapar rollen Customer

```
ALTER ROLE Customer ADD MEMBER Sotis;
```
Lägger till Sotis i rollen

```
DROP ROLE IF EXISTS Customer;
```
Tar Bort rollen Customer

Vilka roller behöver vi?

Vi har tabellerna
music
movies
books
sales + sale_items
people
adresses

Säkerhetsmekanismer (Security Control)

- *Valfria* (Discretionary)
  - Mer direkt styrt
  - För den som har rättighet att skapa tabeller
  - GRANT , DENY & REVOKE
- *Obligatoriska* (Manadatory)
  - Användare & Data i säkerhets nivåer
  - *Unclassified*, *confidential*, *secret* och *top secret*
  - Finns oftast endast i extra säkra system

Exempel för GRANT, DENY & REVOKE

```
DENY ALL ON company_db.* TO Customer;
GRANT SELECT ON company_db.music TO Customer;
GRANT SELECT ON company_db.movies TO Customer;
GRANT SELECT ON company_db.books TO Customer;
GRANT SELECT, INSERT ON company_db.sales TO Customer;
GRANT SELECT, INSERT ON company_db.sale_items TO Customer;
```

Vilka behörigheter ska våra roller ha?

Vi har tabellerna
music
movies
books
sales + sale_items
people
adresses

DML
SELECT
UPDATE
DELETE
INSERT

Behörigheter
GRANT
DENY
REVOKE

[ROLE/USER]
CREATE
DROP

Backup Database

- SQL-injektion
  - Rolig SQL-fråga som bland annat kan tömma hela databasen
  - Klienten & servern skyddar mot detta bäst
- Vi kan använda backups!

```
BACKUP DATABASE company_db TO DISK = 'X:\' + GETDATE() + '.bak';
RESTORE DATABASE company_db TO DISK 'X:\2008-04-20.bak';
```

Transaktioner & Återställning

Transaktion = Följd av operationer som hör ihop som en enhet

LINKÖPINGS UNIVERSITET

Transaktioner

- Tusentals `SELECT`, `UPDATE`, `DELETE`, och `INSERT` per sekund
- Transaktion = Är en följd av såna operationer
- Kan startas med `START TRANSACTION`
- Avslutas med `COMMIT` eller `ROLLBACK`

- Vi abstraherar för att underlätta
  - `SELECT` ➔ Läs
  - `UPDATE`, `DELETE`, och `INSERT` ➔ Skriv

Kommer bli abstrakt och konceptuellt

ACID-transaktion

- (A)tomicity
  - Odelbar = Allt eller inget
- (C)onsistency
  - Före och efter transaktion ska alla integritetsvillkor vara uppfyllda
- (I)solation
  - Vet aldrig om andra transaktioner
- (D)urability
  - Genomförd commit & checkpoint = Permanent

Transaktions operationer sker aldrig samtidigt!

Viktiga delar

Write-ahead logging

Vi djup dyker inte in i hur detta fungerar
Boken ger en bra förklaring. ☺

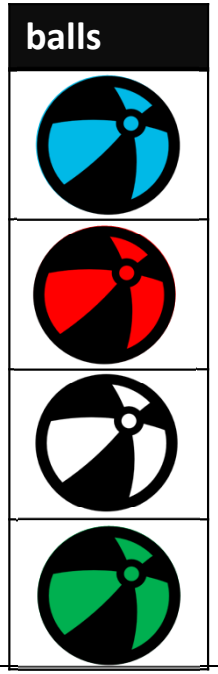
COMMIT, CHECKPOINT & ROLLBACK

- COMMIT (Spara)
  - Transaktion avslutas och skrivs till databasen.
- CHECKPOINT (Spara)
  - Väntar tills alla transaktioner är klara.
  - Sen skrivs loggfilen till hårddisken.
- ROLLBACK (Ångra)
  - Rullar tillbaka till början.

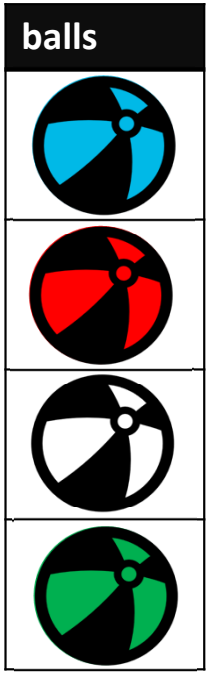
CTRL + S
&
CTRL + Z

Introduktion

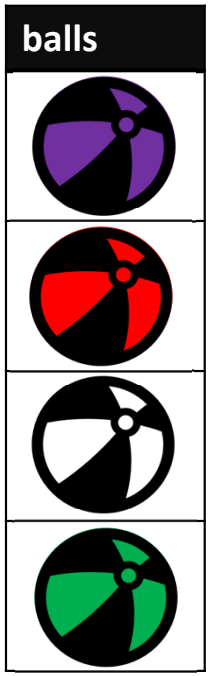
- Happy  och Happier  äger ett Bollbolag



Exempel på Transaktion med Commit



Tid	Operation
1	LÄS
2	LÄS
3	= +
4	SKRIV
5	COMMIT



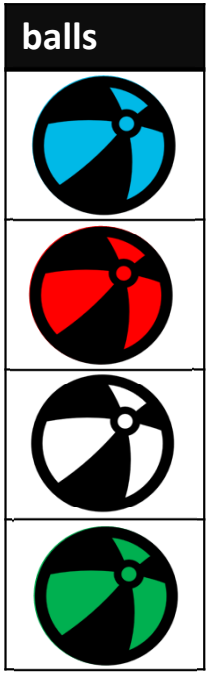
Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

Exempel på Transaktioner med Checkpoint

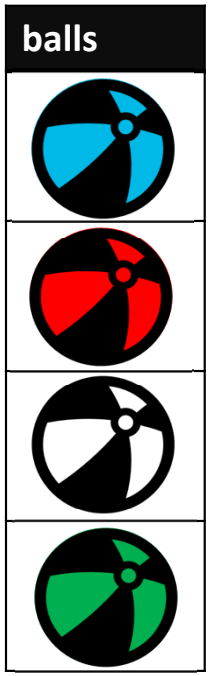
”Tid”	Transaktion 1	Transaktion 2	Transaktion 3	Transaktion 4	Händelse
1	LÄS X		LÄS Z	LÄS J	
2	X + 1	LÄS Y	LÄS K	SKRIV J	
3	SKRIV X	Y + 2	T = K + Z	COMMIT	CHECKPOINT!
4	COMMIT	SKRIV Y	SKRIV T		
5	**KRASCH!**				**Tornado**

Transaktion 1 måste köras om.
Transaktion 2 och 3 måste städas bort.
Transaktion 4 är OK

Exempel på Transaktion med Rollback



Tid	Operation
1	LÄS 
2	LÄS 
3	 =  + 
4	SKRIV 
5	ROLLBACK (or KRASCH)

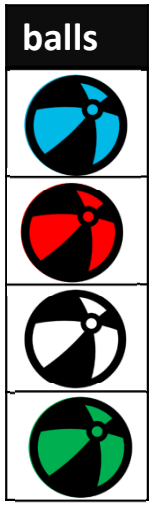


Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

Vad kan gå fel?

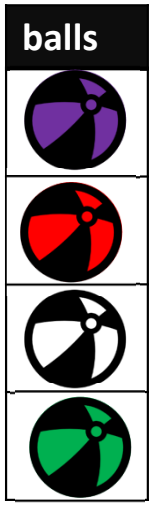
- Flera transaktioner samtidigt!
- Förlorade uppdateringar
- Läsning av smutsig data
- Oupprepbara läsningar

Förlorade uppdateringar (Lost Updates)



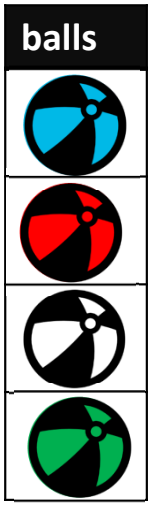
Tid	Operation	Operation
1	LÄS	
2	LÄS	
3	= +	
4		LÄS
5		LÄS
7		= +

Tid	Operation	Operation
8		SKRIV
9	SKRIV	
10		
11		
12		
13		



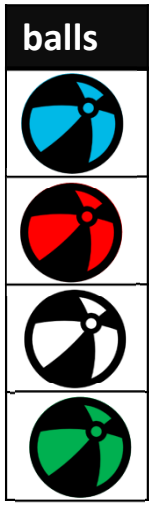
Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

Läsning av smutsig data (Reading Dirty Data)



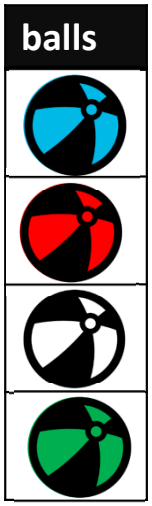
Tid	Operation	Operation
1	LÄS	
2	LÄS	
3	= +	
4	SKRIV	
5		LÄS
7		= +

Tid	Operation	Operation
8		SKRIV
9	ROLLBACK	
10		COMMIT
11		
12		
13		



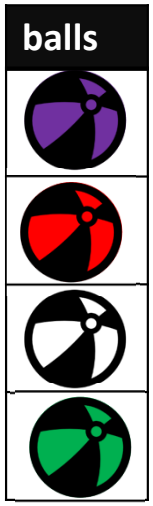
Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

Ouppreppbara läsningar (Non-repeatable read)



Tid	Operation	Operation
1	LÄS	
2		LÄS
3		LÄS
4		= +
5		SKRIV
7		COMMIT

Tid	Operation	Operation
8	LÄS	
9		
10		
11		
12		
13		



Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

Felaktiga summor

Tid	Operation 	Operation 
1	Saldo = 0	
2	LÄS Konto	
3	Saldo = Saldo + Konto	
4		LÄS Konto
5		Konto = Konto – 50
7		SKRIV Konto

Tid	Operation 	Operation 
8		LÄS Lön
9		Konto = Konto + Lön
10		SKRIV Konto
11	LÄS Lön	
12	Konto = Saldo + Lön	
13	Skriv Konto	

Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

Spökposter

Tid	Operation 	Operation 
1	LÄS X	
2		SKRIV X
3	LÄS X	
4		
5		
7		

Transaktion 1	Transaktion 2
SELECT * FROM T WHERE A = 17;	
	INSERT INTO T(A) VALUES (17);
SELECT * FROM T WHERE A = 17;	

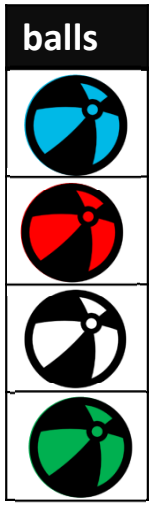
Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

Låsning & Deadlock

- [Simpel] låsning
  - Låser både för läsning och skrivning
- Läs- och skrivlås
  - Låser en tabell baserat på vad man vill göra med den
- Tvåfaslåsning
  - Låser det man behöver, för att sen släppa
  - Har problem med kaskad-rollback
  - Rigorös tvåfaslåsning

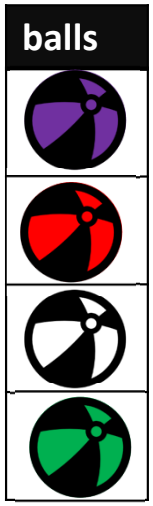
Boken går igenom djupare! ☺
(Kapitel 25)

Förlorade uppdateringar (Lost Updates)



Tid	Operation	Operation
1	LÄS	
2	LÄS	
3	= +	
4		LÄS
5		LÄS
7		= +

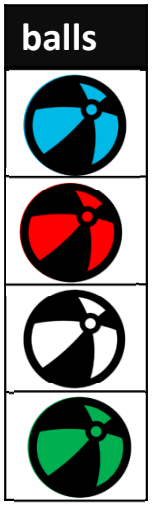
Tid	Operation	Operation
8		SKRIV
9	SKRIV	
10		
11		
12		
13		



Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

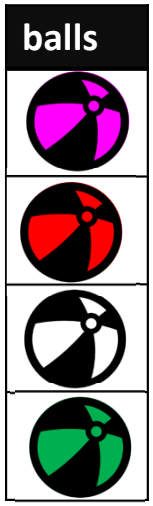
LÅS colors & balls är egentligen två olika operationer

Rigorös tvåfaslåsning



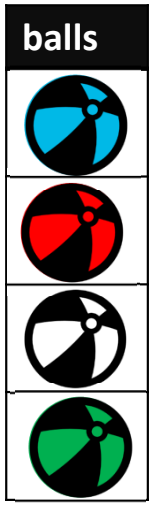
Tid	Operation	Operation
1	LÅS colors & balls	
2		LÅS colors & balls
3	LÄS 	*Väntar…*
4	LÄS 	*Väntar…*
5	 =  + 	*Väntar…*
7	SKRIV 	*Väntar…*

Tid	Operation	Operation
8	COMMIT	**!!!**
9		LÄS 
10		LÄS 
11		 =  + 
12		SKRIV 
13		COMMIT



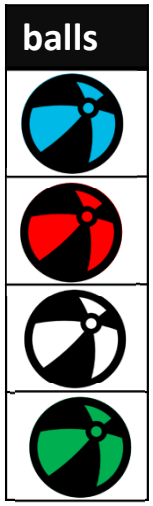
Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

Deadlock



Tid	Operation	Operation
1	LÅS balls	
2		LÅS colors
3	LÅS colors	
4	*Väntar…*	LÅS balls
5	*Väntar…*	*Väntar…*
7	*Väntar…*	*Väntar…*

Tid	Operation	Operation
8	*Väntar…*	*Väntar…*
9	*Väntar…*	*Väntar…*
10	*Väntar…*	*Väntar…*
11	*Väntar…*	*Väntar…*
12	*Väntar…*	*Väntar…*
13	*Väntar…*	*Väntar…*



Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.

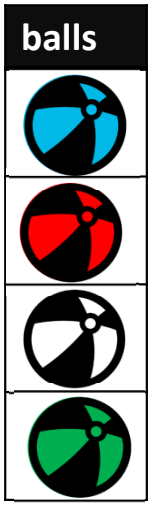
Jonathan Crusoe

Tack för att ni deltog! ☺

www.liu.se

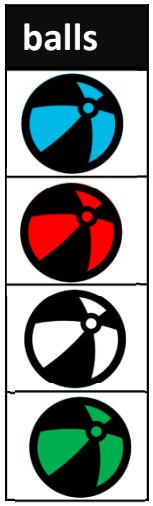
LINKÖPINGS UNIVERSITET

???



Tid	Operation	Operation
1		
2		
3		
4		
5		
7		

Tid	Operation	Operation
8		
9		
10		
11		
12		
13		



Läs = SELECT. **Skriv** = UPDATE, DELETE, INSERT.