Oppgave med transaksjoner og isolation level

## **Vise at en endring ikke er synlig for andre sessjoner før COMMIT**

Disse må evt. utføres før en tar START TRANSACTION;

* set transaction isolation level SERIALIZABLE;
* set transaction isolation level REPEATABLE READ;
* set transaction isolation level READ COMMITTED;
* set transaction isolation level READ UNCOMMITTED
* Slå av autocommit.

Maskingenerert alternativ tekst:
Workbench Preferences 
General Editors 
SQL Editor 
Query Editor 
Object Editors 
SQL Execution 
Administration 
Modeling 
Defaults 
MySQL 
Diagram 
Max. querylength to store in history an bytes): 
Continue SQL script execution on errors ty default) 
w connections use au to comm t mode 
Progress status update Interval an miliseconds): 

Avslutt din connection

Lag 2 hovedvinduer (localhost) mot databasen (disse blir nå med autocommit avslått

Lag et nytt SQL-vindu i hver av dem. Kall dem henholdvis "Conn EEEN" og CONN TOOO"

I Conn EEEN:

* + Sett REPEATABLE READ
  + Start en transaksjon med START TRANSACTION
  + Se på ansatt med ansnr 1
  + Sett Kjønn til B e.l. for ansnr 1 (ikke kjør COMMIT ennå)

Gå til Conn TOOOO

* + Sett REPEATABLE READ
  + Start en transaksjon med START TRANSACTION
  + Se på ansnr 1. Hva står på kjønn?

Gå til Conn EEEN

* + Utfør COMMIT

Gå til Conn TOO igjen:

* + Se på ansnr 1. Hva er kjønn nå?
  + Utfør COMMIT eller ROLLBACK i Conn TOOO

Gjør samme øvelse igjen. Men skriv:

set transaction isolation level READ UNCOMMITTED ;

før START TRANSACTION i Conn TOO

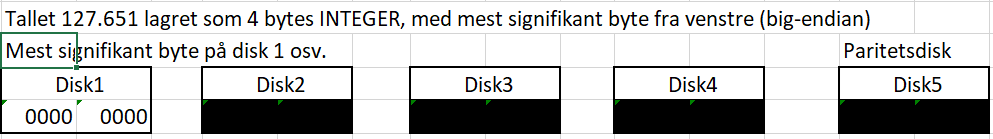
Dersom du får feilmelding om at du ikke kan endre isolation level

når du er i en transaksjon, så gjør en COMMIT eller ROLLBACK først.

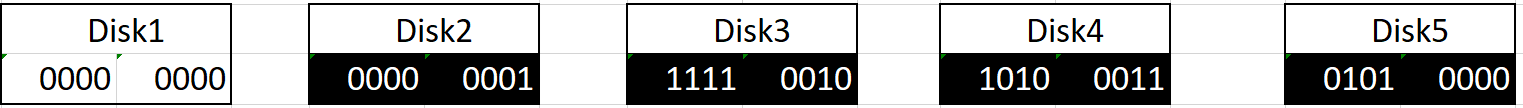
## **Oppgave med RAID**

Du har 5 disker i et RAID 3 oppsett, dvs. striping, med 4 disker + 1 paritetsdisk

1. Vis hvordan tallet 127.651 lagres som en 4 byte INTEGER, med 1 byte på hver disk. Tegn opp de 5 diskene, der mest signifikant byte lagres på DISK 1, nest mest signifikante byte på disk 2 osv. (en slags big-endian lagringsrekkefølge). Tegn opp bitmønsteret (0-ere og 1-ere) på hver byte. Beregn paritets byten på den måten at det til sammen på de 5 diskene alltid skal være et like antall 1’ere på hver bit-plass i byten. Tegn noe lignende som figuren nedenfor.



LØSN:



1. Slett Disk4. Gjennskap disk4 ved hjelp av disk 1,2,3 og 5

## Lag en funksjon i MySQL

1. Lag en funksjon som beregner gjennomsnittspris for en eller flere kategorier i tabellen Hobbyhuset.Vare
2. Test funksjonen

LØSN:

-- En funsjon som beregner gjennomsnitt av en varekategori i Hobbyhuset

DELIMITER $$

CREATE FUNCTION SnittPris(p\_katnr INT)

RETURNS DECIMAL(8, 2)

READS SQL DATA

BEGIN

DECLARE v\_snitt DECIMAL(8, 2);

SELECT AVG(Pris) INTO v\_snitt

FROM Vare

WHERE KatNr = p\_katnr;

RETURN v\_snitt;

END $$

DELIMITER ;

Kalles på denne måten: select snittpris(13);

Slettes med: drop function snittpris;

Se også MySQL innebygde funksjoner: <https://dev.mysql.com/doc/refman/5.7/en/built-in-function-reference.html>

## Lag en procedure i MySQL

1. Lag en procedure med en tekststreng s som input parameter, som returnerer ‘Hallo + <s>’
2. Test proceduren

LØSN:

-- En procedure

DELIMITER $$

CREATE PROCEDURE hobbyhuset.p2 (s char(20))

LANGUAGE SQL

DETERMINISTIC

COMMENT 'Adds "nson" to first and last names in the record.'

BEGIN

SELECT CONCAT('Hello p2 ',s);

END $$

DELIMITER ;

Kalles på denne måten: call p2('morgen');

Slettes med: drop procedure p2;

## Lag en trigger i MySQL

* 1. Lag en trigger som logger prisendringer i Hobbyhuset.Vare til en egen tabell PrisEndringer. Logg både gammel og ny pris.
  2. Test triggeren

LØSN:

-- Trigger som logger prisendringer. Tabellen Prisendring må lages på forhånd.

DELIMITER $$

CREATE TRIGGER Vare\_ARU

AFTER UPDATE ON Vare

FOR EACH ROW

BEGIN

IF (OLD.Pris <> NEW.Pris) THEN

INSERT INTO Prisendring(VNr, Dato, Gammelpris,NyPris)

VALUES (NEW.VNr, CURDATE(), OLD.Pris,NEW.Pris);

END IF;

END $$

Kalles for hver rad som oppdateres i tabellen Vare, etter at raden i Vare er oppdatert

Slettes med: drop trigger Vare\_ARU;

## SQL-injection

* + Lag en veldig enkel php-applikasjon med et input-felt
  + Bruk denne bruker inputen i en dynamisk SQL i php , uten kontroll av av brukerdata
  + Test appen. ved å prøve SQL-injection til f.eks. å slette data i en tabell