1. Repeatable read vs read committed:
   1. Both can only read committed data, but repeatable also prevents other transactions from modifying the data being read until the current read transaction is finished
2. Set of integers where sum = 1000, none are 10, 20, or 30:
   1. Alice’s transaction can produce 1000 if hers finishes first, 1060 if Betty’s finishes first, or 1010 if both Carol and Betty’s transactions finish first. Alice’s query will never read the uncommitted data from Betty’s or Carol’s transactions, and Carol’s transaction can only work successfully if Betty’s finishes before Carol’s starts.
   2. 1030 can be read by Alice’s transaction in **read uncommitted** if Betty’s first two statements occur before the read. This cannot happen in **serializable** because Alice’s transaction would not be able to read uncommitted data and Betty/Carol would not be able to modify the data if Alice’s transaction already started
      1. Possible **read uncommitted** sums: 1000, 1010, 1030, 1060
      2. Possible **serializable** sums: 1000, 1060
3. SQL Scripts:
   1. CREATE VIEW DisneyComedies AS (

SELECT title, length, year

FROM Movie

WHERE (studioName=’Disney’ AND genre=’comedy’));

* 1. CREATE INDEX movieInfo ON Movie (title, year) USING BTREE;

1. CREATE TRIGGER UpdateGpa AFTER INSERT ON CourseGrades

FOR EACH ROW

BEGIN

UPDATE GPA set GPA=(old.GPA\*TotalCredits + Grades\*credits) / (TotalCredits + credits);

UPDATE GPA set TotalCredits=(old.TotalCredits + credits)

END;