Paul Toivonen, Joel Hämäläinen, Amit Yadav
CS-A1150 Databases - Exercise Round 5

1.)
The calculations for the BelongsTo-table are below, with the following changes:
Required pages = 300
One order products = 4
Orders one product is in = 25
As before, the proportion of queries with known order number (Q1) is p1, and the proportion of queries with known product number (Q2) is p2. The proportion of inserts (I) is 1 - p1 - p2. Because there are many more disk pages than the pages required for an index, each result is usually on a different disk page.

When using no index, all disk pages must be checked in all queries. Inserts always require at least disk searches for reading and writing the disk page to be updated.

Using the orderNo Index, the first query requires four disk searches on average to go to the locations in the index, plus one for using the index. Inserts also require two more searches, one for finding the end of the index and one for writing.

Using the productNo Index, the second query requires 25 disk searches to search each product, plus one for using the index. Inserts again require a total of 4..

When using both indexes, both queries are as efficient as before, but inserts require updating both indexes as well, for a total of 6 searches.

The averages can be calculated using the proportions and required disk searches:
No Index: 300p1 + 300p2 + 2(1 - p1 - p2) = 2 + 298p1 + 298p2
orderNo Index: 5p1 + 300p2 + 4(1 - p1 -p2) = 4 + p1 + 296p2
productNo Index: 300p1 + 26p2 + 4(1 - p1 - p2) = 4 + 296p1 + 22p2
Both Indexes: 5p1 + 26p2 + 6(1 - p1 - p2) = 6 - p1 + 20p2

The results have been compiled into the table below.

Action	No Index	orderNo Index	productNo Index	Both Indexes
Q1	300	5	300	5
Q2	300	300	26	26
I	2	4	4	6
Avg	2 + 298p1 + 298p2	4 + p1 + 296p2	4 + 296p1 + 22p2	6 - p1 + 20p2

2.)

a) The first schedule:

It is **not** serializable because no fixed order of execution of transactions T1, T2 and T3 can give the same result as the given schedule.

For example, if the initial value of the balance of account 'A123' is 100, then following the schedule, it'll be 100*1.05 = 105. Because var2 = 100, and after updating it is var2 = 105. And it writes the value after T1 has written. So final value will be 105.

But if we look at order of execution T1, T2 and T3, it'll we (100+100)*1.05 = 210. Or following T2, T1, T3 will give (100*1.05)+100 = 205. Since T3 doesn't change the database, so it's order doesn't matter.

Hence, no order of execution will result in final value of the balance = 105.

The main cause is the overlapping time of read and write of T1 and T2.

b) The second schedule

It is serializable. The order of execution T1, T3 and T2 give same result as this.

For example, if initial balance is 100, then after the given schedule it's final value will be (100+100)*1.05 = 210. And T3 will output 200.

And following the schedule T1, T3, T1 will also result in final value = (100+100)*1.05 = 210. And T3 will also output value 200.

So it's serializable.

3.)

a)
```
CREATE TRIGGER T1
AFTER INSERT on Courses
FOR EACH ROW
WHEN New.credits>10
begin
        UPDATE courses SET credits=10 where code=New.code;
end
```

b)
```
create trigger T2
after update of grade on Grades
for each row
when New.grade<Old.grade
begin
        update Grades
        set grade=old.grade
        where studentID=New.studentID;
```

```
end;
```

c)
```
create trigger T3
after update of credits on Courses
for each row
when New.credits>2+Old.credits
begin
        update Courses
        set credits = Old.credits+2
        where code=Old.code;
end;
```

d)

i) insert into Courses Values ('test', 'Test Name', 15);
ii) insert into Courses Values ('test2', 'Test Name2', 9);
iii) update Grades set grade=6 where studentID=112233;
iv) update Grades set grade=3 where studentID=112233;
v) update Courses set credits=10 where code='CS-A1111';
vi) update Courses set credits=10;

Before executing:

Courses:

code	name	credits
CS-A1111	Basic Course in Programming Y1	5
CS-A1120	Programming 2	5

Grades:

studentID	courseCode	date	grade
112233	CS-A1111	2017-12-05	3
112233	CS-A1120	2018-12-19	1
224411	CS-A1111	2018-12-12	4

After executing:

Courses:

code	name	credits
CS-A1111	Basic Course in Programming Y1	9
CS-A1120	Programming 2	7
test	Test Name	10
test2	Test Name2	10

Grades:

studentID	courseCode	date	grade
112233	CS-A1111	2017-12-05	6
112233	CS-A1120	2018-12-19	6

4. & 5.)
These exercises are in a separate python program.
Results from example execution:

```
python SQLPython.py
Database created
The year is 1963

Input command: C = Add Company / A = Add Album / S = Search / Q = Quit
C
Insert Company: Write company info, separate fields with spaces
Universal USA universal.com

Inserted Universal

Insert command: C = Add Company / A = Add Album / S = Search / Q = Quit
C
Insert Company: Write company info, separate fields with spaces
Sony Japan sony.com

Inserted Sony

Insert command: C = Add Company / A = Add Album / S = Search / Q = Quit
C
Insert Company: Write company info, separate fields with spaces
Warner USA warner.com

Inserted Warner

Insert command: C = Add Company / A = Add Album / S = Search / Q = Quit
A
Insert Album: Write album info, separate fields with spaces
Clapton Warner 2010 46 rock

Inserted Clapton

Input command: C = Add Company / A = Add Album / S = Search / Q = Quit
A
Insert Album: Write album info, separate fields with spaces
Demon_Days Warner 2005 50 rock
```

```
Inserted Demon_Days

Insert command: C = Add Company / A = Add Album / S = Search / Q = Quit
A
Insert Album: Write album info, separate fields with spaces
x Warner 2014 70 pop

Inserted x

Insert command: C = Add Company / A = Add Album / S = Search / Q = Quit
A
Insert Album: Write album info, separate fields with spaces
Blood Warner 2015 42 folk

Inserted Blood

Insert command: C = Add Company / A = Add Album / S = Search / Q = Quit
A
Insert Album: Write album info, separate fields with spaces
Timeless Sony 1969 31 jazz

Inserted Timeless

Insert command: C = Add Company / A = Add Album / S = Search / Q = Quit
A
Insert Album: Write album info, separate fields with spaces
Best_of_Mozart Universal 2018 85 classical

Inserted Best_of_Mozart

Insert command: C = Add Company / A = Add Album / S = Search / Q = Quit
S
Search Company: Write company name
Warner

Albums published by Warner
Clapton 2010 rock
Demon_Days 2005 rock
x 2014 pop
Blood 2015 folk

Insert command: C = Add Company / A = Add Album / S = Search / Q = Quit
Q
EXITING
```