CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
DIT 033	Data Management			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
245		2022-03-14	10	~~AVG~~ 5

* I confirm that I've no mobile or other similar electronic equipment available during the examination.
Jag intygar att jag inte har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under eximinationen.

Solved task Behandlade uppgifter No/nr		Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylles av lärare.
1	X	23.5	
2	X	19	
3	X	12	
4	X	19	
5	X	18	
6	X	4	
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points Summa poäng på tentamen	95.5		

3

Q1.1 A Cartesian Product is an operation that takes every row from one table and pairs it with every row of the other table, producing a table with n*m rows (n is nr. of rows in table1 and m in table2), the resulting table has attributes from both tables.

Example: U_STUDENT

name	s_id
Kim	1
Noah	2
John	3

GRADE

student	course	grade
1	DIT_033	4
2	DIT_034	5
3	DIT_033	3

U_STUDENT x GRADES gives:

name	s_id	student	course	grade
Kim	1	1	DIT_033	4
Kim	1	2	DIT_034	5
Kim	1	3	DIT_033	3
Noah	2	1	DIT_033	4
Noah	2	2	DIT_034	5
Noah	2	3	DIT_033	3
John	3	1	DIT_033	4
John	3	2	DIT_034	5
John	3	3	DIT_033	3

3

Q1.1 continuation

The difference between a join and a cartesian product is that a join first does a cartesian Product operation on tables and then preforms a SELECT operation based on a specified condition

In RA a join can be written as:

$\sigma_{condition}(table\_one \times table\_two)$ -0.5 incorrect example of how join can be specified as cartesian product

Ex: $\sigma_{s\_id=student}(U\_STUDENT \times GRADE)$

4.5/5

Q1.2 Normalization is the process of reforming a database so that it becomes normal.

Consider RM: REVIEW(user, place, comment, stars_given, stars_meaning).

REVIEW is not 3NF since there is a relational dependency stars_given → stars_meaning (so 4 stars would for example imply "great"), which is a so called transient dependency and must be removed, as a non-key attrib must depend on a key attribute. Besides that, criteria for 2NF must also be fulfilled.

2

3

Q1.3 The indexes are used to optimize SELECT operation on a tables attribute, so it does not need to search the entire table.

(3)

Advantage: SELECT becomes much faster (O(log n) instead of O(n))

Disadvantage: storing an index takes extra memory space.

Q1.4 Well-formedness means adherance to syntax rules of a given file format. A valid document must be well-formed, but the reverse does not need to be true

XML rules: 1. every tag that is opened must be closed

Ex: either <a> or <a />

2. There can only be one root tag, after it is closed, there can be no further tags, not even comments.

(4)

3

Q1.5 A "non-repeatable read" happens when two reads in a transaction return different values of attributes for a given entry in a table, because it was modified by a write outside of the given transaction.

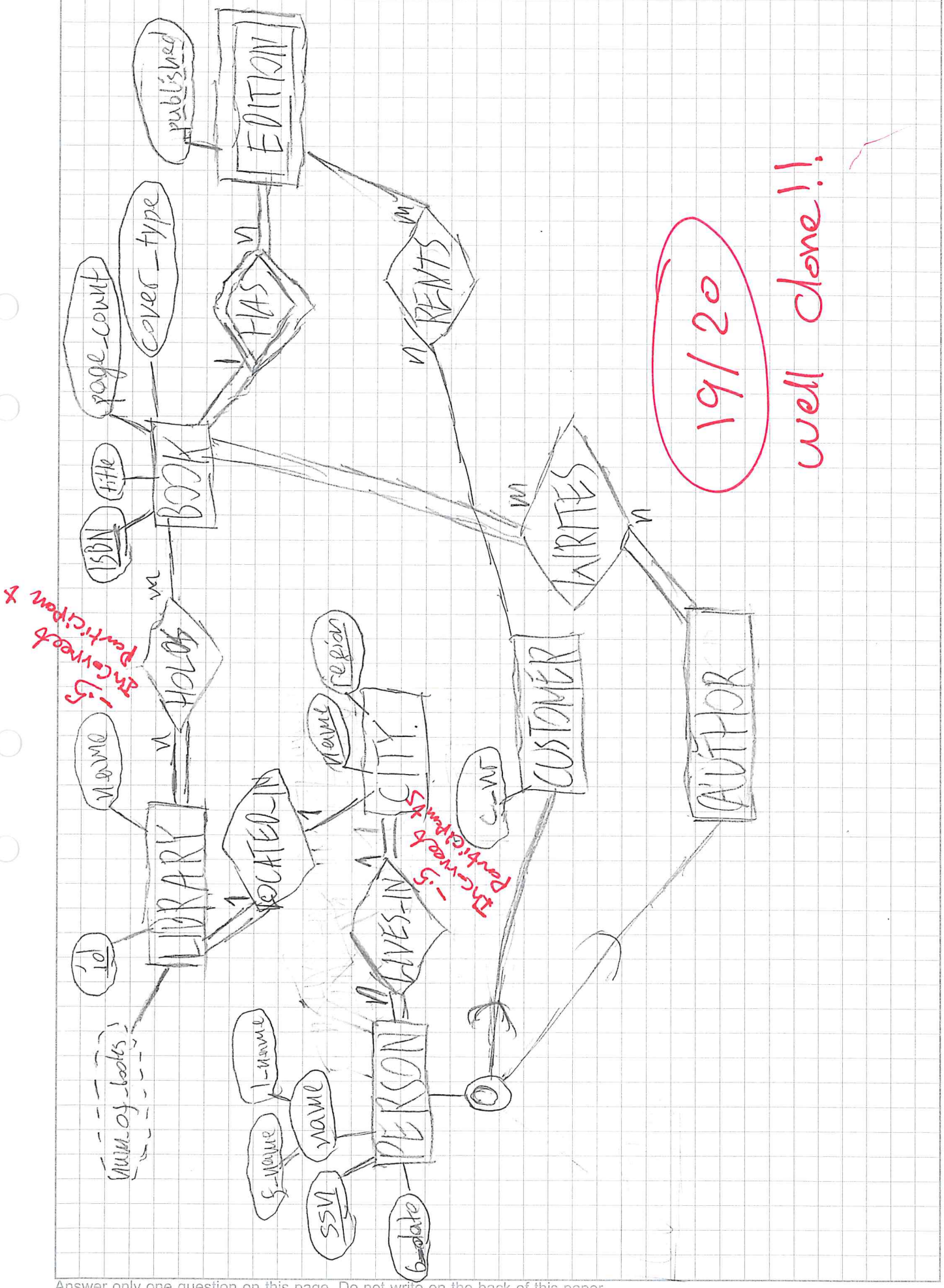
5

isolation level	dirty read	non-repeatable read	phantom
READ UNCOMMITTED	yes	yes	yes
READ COMMITTED	no	yes	yes
REPEATABLE READ	no	no	yes
SERIALIZED	no	no	no

Q1.6 Replication is useful as it improves scalability of a DB (load distribution) and availability (eliminates single-point failure). 2 types of replication are: primary/primary and primary/secondary. Replication is copying data across multiple nodes thus making redundancies, whereas sharding is dividing data across nodes without making copies.

5

3



19/20

well done!!

3

LIBRARY(id, name, city_name, city_region, num_of_books)
{city_name, city_region} → {CITY.name, CITY.region}

CITY(name, region)

PERSON(ssn, f_name, l_name, b_date, city_name, city_reg
{city_name, city_region} → {CITY.name, CITY.region}

CUSTOMER(ssn, c_nr)
{ssn} → {PERSON.ssn}

AUTHOR(ssn)
{ssn} → {PERSON.ssn}

BOOK(ISBN, title, page_count, cover_type)

LIB_HOLDS_BOOK(library, book)
{library} → {LIBRARY.id}
{book} → {BOOK.ISBN}

AUTHOR_WRITES_BOOK(author, book)
{author} → {AUTHOR.ssn}
{book} → {BOOK.ISBN}

3

EDITION(published, book)

{book} → {BOOK.ISBN}

CUSTOMER_RENTS_EDITION(customer, book, published

{customer} → {CUSTOMER.ssn}

{book, published} → {EDITION.book, EDITION.published}

12/12

well done!

3

Q4.1 $\Pi_{since}(\sigma_{address="kavi1/gatan\ 11" \wedge first\_name="John" \wedge last\_name="Smith" \wedge pet="lulu"}(PET\_LIVES\_WITH\_AT))$ 5

Q4.2 $DOG \leftarrow \Pi_{name}(\sigma_{type="dog"}(PET))$

$\Pi_{name, firstname, last\_name, birthday}(HUMAN * PET\_OWNER \bowtie_{pet=name} DOG)$ 5

Q4.3 $V\_ADDRESS \leftarrow \Pi_{address, pet}(PET\_LIVES\_WITH\_AT)$

$address\ \mathcal{F}COUNT\ pet\ (V\_ADDRESS)$ 5

Q4.4

$PO \leftarrow PET\_OWNER$

$\Pi_{HUMAN.first\_name, HUMAN.last\_name, pet}(HUMAN \bowtie PO)$ 4

↑ join condition?

Q5.1 SELECT * FROM HUMAN ORDER BY age DESC; 4

Q5.2 SELECT H.address, year_built, pet FROM HOUSE AS H LEFT JOIN (SELECT DISTINCT address, pet FROM PET_LIVES_WITH_AT) AS P ON H.address = P.address; 4

Q5.3 SELECT AVG(age) FROM PET WHERE type = "cat"; 5

Q5.4 SELECT name FROM PET WHERE name IN (SELECT first_name FROM HUMAN); 5

Q6.1 Returns value of attribute student.id and _id for all students with the name "James Lewis". The output is prettified. 2

Q6.2 Sets the value of student.enrolled to false for all students with the name "James Lewis". If the value of student.enrolled was already false, it is not updated. 2