CHALMERS

7988870

FÖRSÄTTSBLAD TENTAMEN / EXAMINATION COVER

Kurskod / Course code:	Provkod / Test code:	Tentamensdatum / Examination date:
DIT034	1011	2025-08-20
Anonymt kodnummer / Anonymous code number:	0014-SHR	
Kursnamn / Course name:	Systematisk Datahantering	

Ifylles av student / To be completed by the student

Behandlade uppgifter. Sätt kryss (X) / Solved assignment. Put an X.:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
☒	☒	☒	☒	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Antal inlämnade svarsblad. Sätt kryss (X) i rutorna / Number of submitted answer sheets. Put a/an X in boxes.

0	10	20	30	40	50	60	70	80	90	+	1	2	3	4	5	6	7	8	9
☐	☒	☐	☐	☐	☐	☐	☐	☐	☐		☐	☐	☐	☐	☐	☐	☐	☐	☐

Ifylles av lärare / To be completed by the examiner

Poäng på uppgifter / Points per question										
1	2	3	4	5	6	7	8	9	10	Bonus
11	12	13	14	15	16	17	18	19	20	Total
										,
										Heltal / integer *0,5*

Datainläsning

Totalpoäng / Total points										
10	20	30	40	50	60	70	80	90	100	200
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
+	0,5	1	2	3	4	5	6	7	8	9
	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Betygskala/ Grade scale	Betyg / Grade			
	U	3	4	5
TH	☐	☐	☐	☐
	U	G	VG	
UV/UG	☐	☐	☐	

7988870

Lärarens kommentarer

3

Q1.1) A weak entity is an entity that depends on another entity for existing. It doesn't have a primary key of its own. It is special as instead it has a partial key. It's primary key is composed of the partial key and foreign key refering to the ~~table~~ primary key of another table due to which this entity exists. We use weak entity when the entity is fully dependent on something else ~~for so~~ if a doesn't exist & would'nt as well. for example:- hotel and its room. ~~Ho R~~ Hotel rooms are a weak entity as they won't exist if the hotel entity doesn't exist. Hence each hotel room will have a partial key ~~and wi~~ which will tell its number however that together with the hotel name will be uniquely identified.

+5

3

Q1.2) when two transactions are happening in parallel there are chances of dirty read if the level of database isolation is READ UNCOMMITED. As User 1 reads a value of x (eg(20)) and then changes it to 30. ~~Then~~ It's still not committed however another user also reads x which was just made so by another user. Now user 1 ROLLBACK and doest commit or changes the value of x again and then commits. While User 2 read the value of x as 30 and did its transaction ~~ete~~ according and then commited. This will cause dirty read as User 2 used a value that was never committed.

+5

Read (x)
Read (y)
X = 20

3

1.3) ~~Q3~~ Slow queries can be improved by use of views, Indices and Query Rewriting.

Query Rewriting is a process in which we write the query again in a different way which gives the same result but more efficiently. There is heuristic and cost based query rewriting. In heuristic we use the rules ~~of~~ → for rewriting more efficient queiries. For example: $\pi_{LIST1}(\pi_{LIST2}(EMPLOYEE)) \equiv \pi_{LIST1}(EMPLOYEE$

↳ This one is faster but this rule is applicable iff List1 is a subset of List2.

+4

So both ~~are~~ will give same result but one is faster, the the other.

The cost based query rewriting is based on the data itself and the database system we are using.

In view the query that is used many time has its results stored on disk. So the query doesn't have to ~~be~~ run completely again and find data.

This is materialized views

3

CHALMERS | Anonymous code / Anonym kod: DITO 34-0014-SHR | Points for question / Poäng på uppgiften | Consecutive page no. / Löpande sid nr: 4 | Question no. / Uppgift nr: 1.4

1.4) In map/reduce a large amount of data is divided into sharads. ~~Each sharad is then~~ Then the data in sharad is made into key:value pair which is mapping. Key is the data and value it the number of times it appears. ~~Then it~~ ~~At first with each piece of data~~

This is done with reduce

+2

Example missing

3

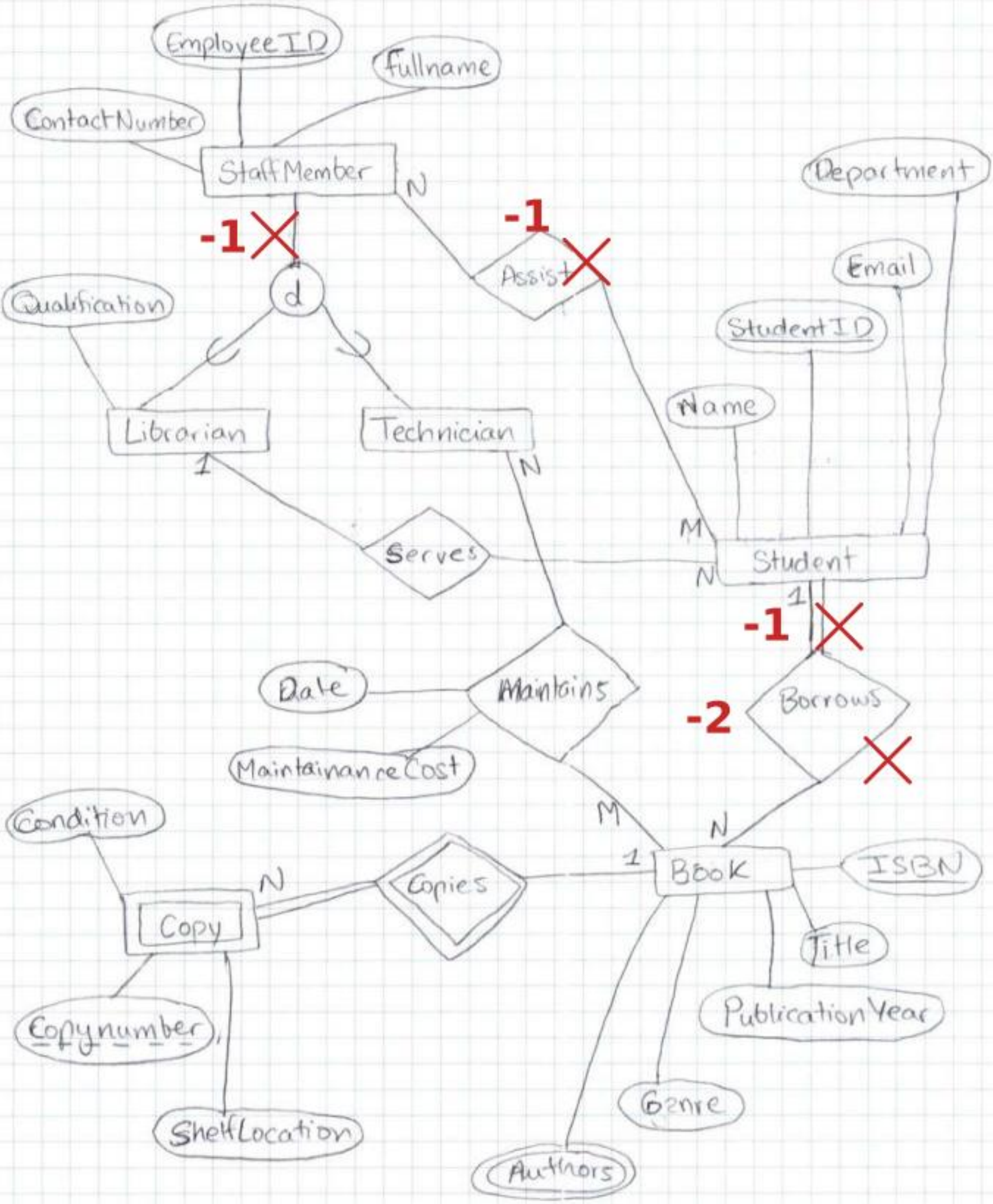
Assumptions made:-

- its written "staff member is either Librarian ..., or Technician" which I assume is to tell ~~for~~ that & it's distinct. It's not mentioned that they must be either of them hence assuming its partial participation.

- ~~Assist relationship~~
- "Book can be borrowed by multiple students." at different time" ~~matte~~ gives impression that at any given time the relationship ~~as~~ is 1:N as ~~stud~~ book can't be borrowed by many students at same time.

This should not impact the cardinality.. still M:N

3


EmployeeID
Fullname
ContactNumber
StaffMember
N
-1
-1
Assist
Department
Email
StudentID
Name
d
Qualification
Librarian
Technician
1
N
M
Serves
Student
N
1
-1
Date
Maintains
-2
Borrows
MaintainanceCost
Condition
M
N
1
Book
ISBN
N
Copies
Copy
Title
PublicationYear
Copynumber
Genre
ShelfLocation
Authors

3

ACTOR (ActorID, Name, Nationality) ✓ +4

~~Lead~~
LEADACTOR (ActorID)
ActorID → ACTOR.ActorID ✓

PREVIOUSFILMS (Film, Actor)
~~Actor → ActorID~~
Actor → LEAD_ACTOR.ActorID ✓

+4
SUPPORTING_ACTOR (ActorID, ExperienceYears)
ActorID → ACTOR.ActorID

FILMCREW (CrewID, Name, Role) ✓

DIRECTOR (CrewID, AwardCount)
CrewID → ~~CrewID.~~ FILMCREW.CrewID ✓ +4

€
CINEMATOGRAPHER (CrewID, CameraStyle) ✓
CrewID → FILMCREW.CrewID

~~SHOOTS (CrewID, Film~~
SHOOTS (Crew, Film, SceneSetup) +4
Crew → CINEMATOGRAPHER.CrewID ✓
Film → FILM.FilmID

PLAYS (Crew, Film)
Crew → FILMCREW.CrewID
Film → FILM.FilmID ✓

ACTSIN (~~Crew~~ Film, Actor) +4
Film → FILM.FilmID ✓
Actor → ACTOR.ActorID

FILM (FilmID, Title, ReleaseYear) ✓

CHALMERS | Anonymous code / Anonym kod: DIT034 0014 SHR | Points for question / Poäng på uppgiften (to be filled in by teacher) | Consecutive page no. / Löpande sid nr: 8 | Question no. / Uppgift nr: 4

4.1) $\pi_{genre}(MOVIE)$ **+5**

4.2) $\pi_{completion \times duration}(STREAM \bowtie movie = title\ MOVIE)$

Missing stream id

+4

4.3) ~~User.username Γ COUNT{movie}(~~ ~~$\sigma_{reg\_date = date}$~~

$(USER \bowtie username = user\ STREAM))$

+4

4.4)

Use projection here to remove ambiguity

$WithoutMark \leftarrow \sigma(name <> "Mark\ Ruffalo"(ACTOR \bowtie movie = title\ MOVIE))$

$\pi_{Title, genre}(\sigma_{username = "Phillip"}(USER \bowtie username = user\ STREAM$

+3 $\bowtie^{STREAM.}_{movie = title}\ Without\_MARK))$

3

5.1)
```
SELECT *
FROM USER
WHERE reg_date = "05-04-2024"
ORDER BY username;
```
+5

5.2)
```
SELECT SUM(completion x duration), username
FROM USER, STREAM, MOVIE
WHERE User = username AND movie = title
GROUP BY username
HAVING quata > SUM(completion x duration)
```
+3

3

5.3) SELECT DISTINCT username, ~~quota~~, ~~reg~~date, ~~DISTINCT~~ ~~movies~~
FROM USERS LEFT OUTER JOIN STREAM ON
~~u~~username = user +2

5.4) SELECT name, COUNT(*)
FROM ACTORS
GROUP BY name
~~HAVING COUNT > COUNT(*)~~

CHALMERS			
			Question no. Uppgift nr

Question	Your score / Max
Theory and Understanding	14 / 20
EER Diagrams	15 / 20
Mapping an EER model	20 / 20
Relational Algebra	16 / 20
SQL	10 / 20
Total	**75 / 100**