emails(idm, from, to, subject, body)
attachments(ida, idm, filename, type)

Write SQL queries

1. For every email return two columns: subject, number of attachments
2. Return subject, body for every email that has more than 2 attachments
3. Return subject of all emails that have more attachments than mail with idm=
4. Return filenames of all attachments that belong to emails sent by user XY
   (field from)

Task consists of ten connected parts. All should be done in correct order. Time 20 minutes.

1. Log as user ADM with password AS10
2. Connect to database TEST
3. Create user HEAD with password XYZ10
4. Give user HEAD rights to delete records from table CARS(CAR_ID, REG_NUM)
5. Create table MODELS with fields MODEL_ID and MODEL_NAME.
6. Modify both tables to ensure that every car has information about its model
7. Ensure that it is impossible to remove model when there are cars of this model
8. Add table OWNERS and connect it with table CARS (every owner may have multiple cars)
9. Ensure that it is impossible to add car without its owner
10. Give a new user CAROWNER rights to modify table CARS

Create a project of the database that stores information about a conference. The conference consists of sessions. On every session several papers are presented. Each paper has authors. Author may write more than one paper. The conference has participants. Participant may be author of a paper but it is not necessary. There are different types of participants as: VIPs, authors, students etc. Every participant belongs to one type. For each type of participant there is a different conference fee. There are several facultative attractions during the conference (like: excursion or banquet) in which participants may take part. Every participant may take part in several attractions.

Create logical ERD for the database storing all information mentioned above. Basing on th ERD develop SQL schema of the database (set of CREATE TABLE statements). Underli primary keys and draw arrows between foreign keys and corresponding primary keys.

Databases Exam
5325514 [redacted]
name .....

The student individually indicates the correctness of every answer according to the introductory part. There are three possi
- mark the answer as a correct one – putting sign **1 (one)** in the box before the answer
- mark the answer as a false one – putting sign **0 (zero)** in the box before the answer
- leave unanswered - putting sign **X** in the box before the answer

To change previously checked answer the student must smear the box, sign it, draw a new box and put there a new answe
Student gets 2 points for correct answer and –3 points for incorrect answer. To pass the exam, two conditions have to b
fulfilled simultaneously: 20 correct answers and positive number of points.

For relations: model(idm, make, type), car(idc, idm, regnum, year) and equipment(ide, idc, name)
he query: "Find regnums of cars of make FIAT having equipment named ABS" may be realized with:

$\Pi_{regnum}(\delta_{make='FIAT' \text{ and } name='ABS'}(model \underset{model.idm=car.idm}{\triangleright\triangleleft} car \underset{car.idc=equipment.idc}{\triangleright\triangleleft} \Pi_{name}(equipment)))$

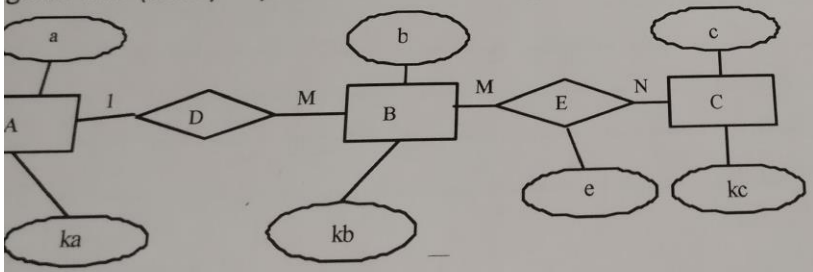
$\Pi_{regnum}(\delta_{make='FIAT'}(model) \underset{model.idm=car.idm}{\triangleright\triangleleft} \Pi_{regnum,idc,idm}(car) \underset{car.idc=equipment.idc}{\triangleright\triangleleft} \delta_{name='ABS'}(equipment))$

$\Pi_{regnum,idc}(\delta_{make='FIAT'}(model) \underset{model.idm=car.idm}{\triangleright\triangleleft} car) \underset{car.idc=equipment.idc}{\triangleright\triangleleft} \delta_{name='ABS'}(equipment)$

$\delta_{make='FIAT'}(model) \underset{model.idm=car.idm}{\triangleright\triangleleft} \Pi_{regnum}(car) \underset{car.idc=equipment.idc}{\triangleright\triangleleft} \delta_{name='ABS'}(equipment)$

$I_{regnum}(\delta_{make='FIAT'}(model \underset{model.idm=car.idm}{\triangleright\triangleleft} car \underset{car.idc=equipment.idc}{\triangleright\triangleleft} \delta_{name='ABS'}(equipment)))$

given ERD (entity keys have names starting with "k")



below statements are true:

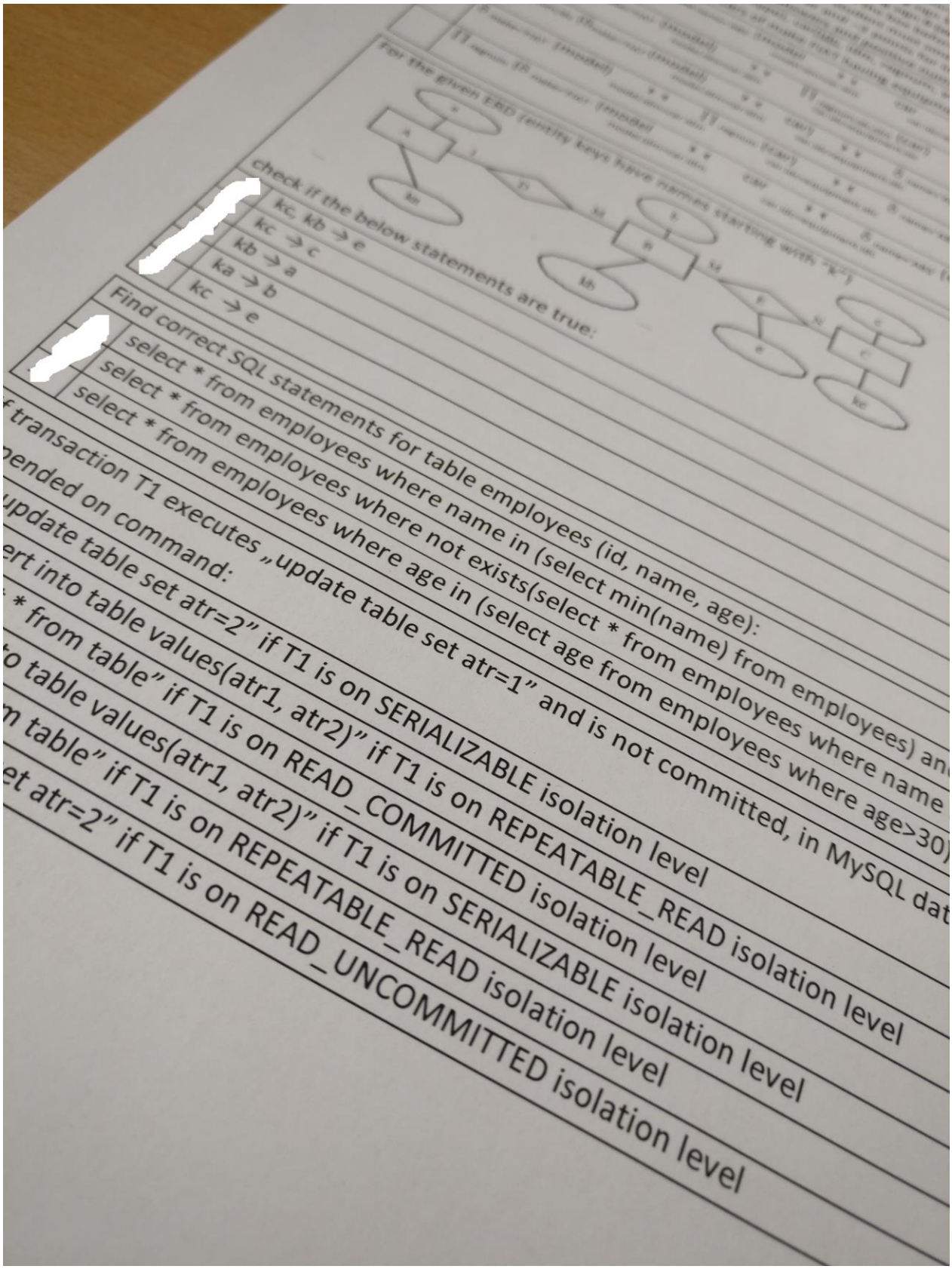
→ e

L statements for table employees (id, name, age):

m employees where name in (select min(name) from employees) and age<30;

n employees where not exists(select * from employees where name like 'K%');

[illegible] age in (select age from employees where age>30);



For the given ERD (entity keys have names starting with "k")

check if the below statements are true:

kc, kb → e
kc → c
kb → a
ka → b
kc → e

Find correct SQL statements for table employees (id, name, age):

select * from employees where name in (select min(name) from employees where name
select * from employees where not exists(select * from employees where age>30)
select * from employees where age in (select age from employees where age>30)

f transaction T1 executes „update table set atr=1" and is not committed, in MySQL dat
pended on command:

update table set atr=2" if T1 is on SERIALIZABLE isolation level
ert into table values(atr1, atr2)" if T1 is on REPEATABLE_READ isolation level
* from table" if T1 is on READ_COMMITTED isolation level
o table values(atr1, atr2)" if T1 is on SERIALIZABLE isolation level
m table" if T1 is on READ_READ isolation level
et atr=2" if T1 is on REPEATABLE_READ isolation level
et atr=2" if T1 is on READ_UNCOMMITTED isolation level

Judge correctness of statements:

- if relation is not in third normal form it cannot be in second normal form
- Non atomic attributes are possible if relation is in second normal form

Having the following set of functional dependencies (AB → C, B → F, B → C, C → F) we can say that (keys underlined):

- Schema (B, C, D) is in third normal form
- Schema (A, B, E) is in second normal form
- Schema (A, B, C, D) is in second normal form
- Schema (B, D, E, F) is in third normal form

For relation r(R), where R={ABC} and relation s(S), where S={CD}.

$$r = \begin{array}{ccc} A & B & C \\ a1 & b1 & c1 \\ a1 & b2 & c2 \\ a2 & b2 & c2 \end{array} \quad s = \begin{array}{cc} C & D \\ c1 & d1 \\ c1 & d2 \end{array}$$

Evaluate if the equations below are true (⋈ - means natural join) :

- $\pi_{CD}(r \bowtie s) = \pi_C(r) \bowtie \pi_{CD}(s)$
- $\pi_{ABC}(r \bowtie s) = r$
- $\sigma_{D=d1}(s) \bowtie r = s \bowtie r$
- $\sigma_{A=a1 \wedge D=d2}(r \bowtie s) = \sigma_{A=b2}(r) \bowtie \sigma_{D=d2}(s)$
- $\pi_{AB}(\sigma_{A=a1}(r \bowtie s)) = \sigma_{A=a1 \wedge B=b1}(\pi_{AB}(r))$
- $\sigma_{C=c1}(r \bowtie s) = \sigma_{C=c1}(r) \bowtie s$

are two tables in the database:

(idA ,description)

idB, idA, description)

ecution of the statement:

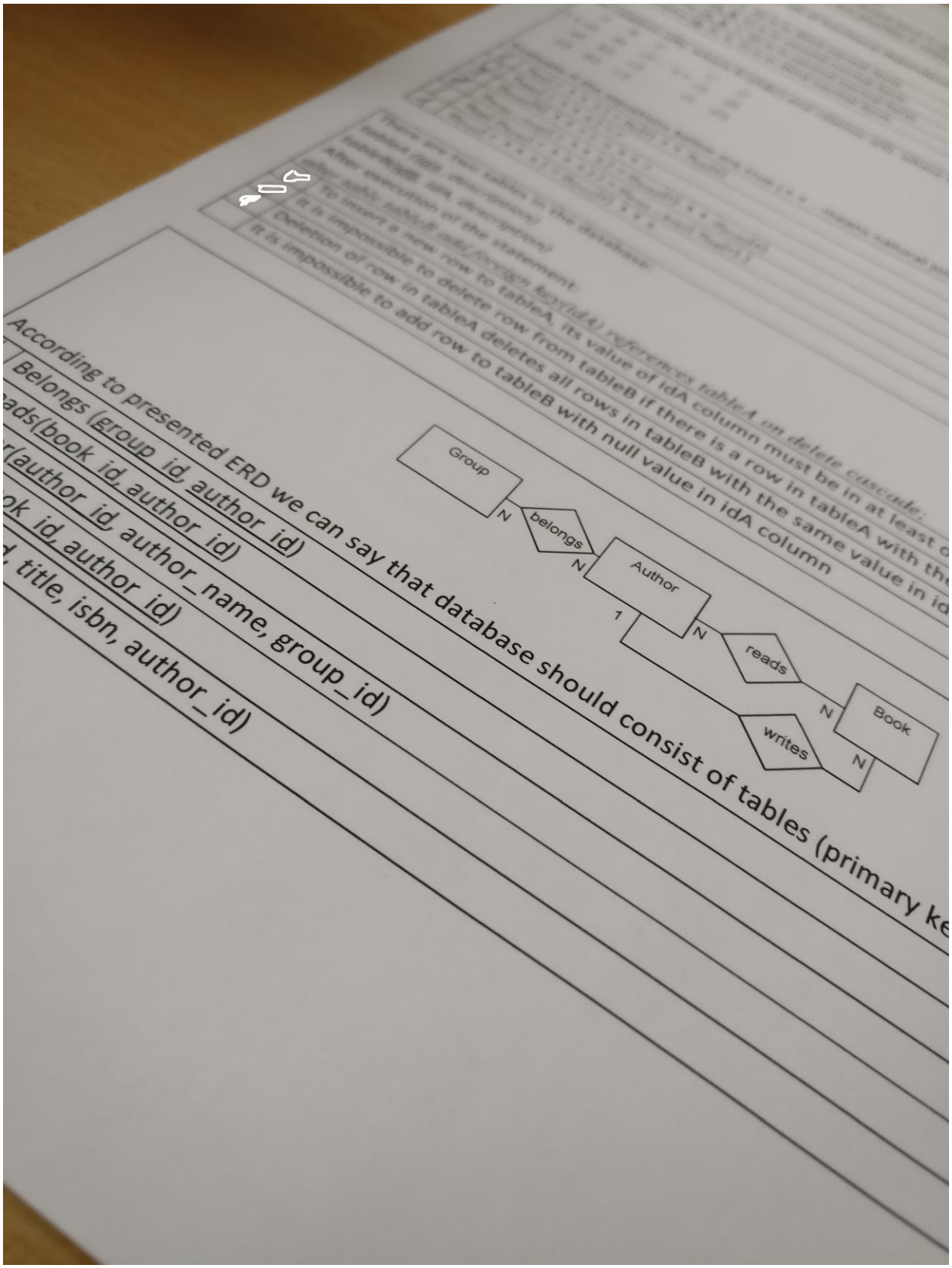
e tableB add foreign key(idA) references tableA on delete cascade;

sert a new row to tableA, its value of idA column must be in at least one

possible to delete row from tableB if there is a row in tableA with the

n of row in tableA deletes all rows in tableB with the same value in

ossible to add row to tableB with null value in idA column


Group
N
belongs
N
Author
1
N
reads
N
Book
writes
N
According to presented ERD we can say that database should consist of tables (primary ke
Belongs (group_id, author_id)
ads(book_id, author_id)
(author_id, author_name, group_id)
ok_id, author_id)
, title, isbn, author_id)