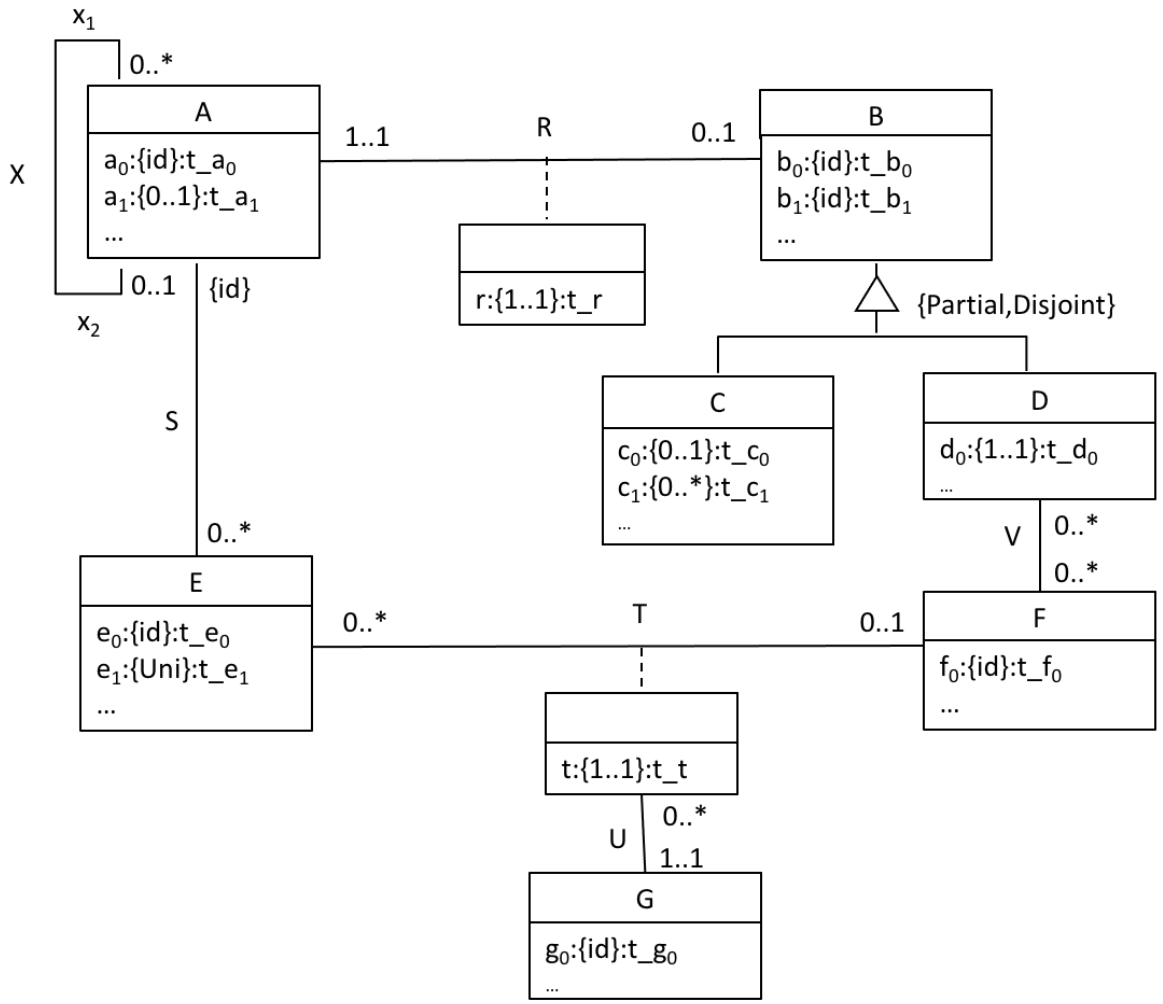
DATABASES AND INFORMATION SYSTEMS (15/1/21)

1. **(1.75 points)** Perform the logical design of the following UML class diagram in order to obtain the equivalent set of relations in the relational model. It is not necessary to indicate the data types in the relations. Those constraints that cannot be expressed graphically can be written in natural language (English or Spanish).



2. **(0.5 points)** Consider the following relational schema:

R(A: integer, B: text, C: set of integers, D: text, E: text, F: text, G: text)
PK: {A, B}
NNV: {C, D, E, F, G}

From the functional dependencies shown below, transform the relation to a set of relations in third normal form.

{B} → {D} {D} → {E} {E} → {F}

3. **(1.75 points**) Design a UML class diagram for the following information system. Those constraints that cannot be expressed graphically can be written in natural language (English or Spanish).

We want to launch an app store for a new mobile device that is going to be released. The information needed is the following.

Users need to register in order to access the store. The information required for the registration process is the user ID, name, email (which is unique), and date of registration. Applications are uploaded by users and, after an approval process, are available for download in the store. The applications have a name (which is unique), a description, one or more categories, date of registration and type of license. Once an application is created, it is assigned an internal code that identifies it. The categories have a code and a description, and each application must be classified in at least one of them.

Once the application is approved, the approval date is saved and, from then on, its owner can create versions, which are the ones that users will download. Each version is identified by a unique number within each application. Each version can have a name, a price proposed by the user, a release date and some observations, where other characteristics or restrictions of the version can be indicated.

Each time a user downloads a version of the application, the date of the download is saved. Users can download it many times, although they will only be charged the first time (assuming it is not free). Of the downloaded versions, the user can add one (and only one) rating between 1 and 5. On the other hand, the user can also add as many comments as he/she wants about the applications, saving for each of the comments the instant (date and time), the message, and in case the comment is a response to another (from the same application), it will also be saved to which comment is responding. A comment is identified with the identifier of the user who creates it and the instant of creation.

Normally, the amount that the user pays for the versions of the applications is the one proposed by the creator, although periodically the store launches promotions to reduce the actual price of some downloads. It is therefore important to keep a history of the prices of each version in order to know the price according to the time of purchase. In this history the start date, the end date and the final price to be paid are needed. For any particular moment there can only be one price for each version.

DATABASES AND INFORMATION SYSTEMS (15/1/21)

This questionnaire consists of 10 questions; for each one four answers are proposed, of which **only one** is correct. The answer should be included in the answer sheet provided separately. The maximum score for the questionnaire is 1 point. The score obtained is calculated with the formula **(Correct_answers** – **Wrong answers/3)** × **0,1**.

1 Select the **TRUE** statement regarding the conceptual scheme
   a. Describes an information system from an organizational point of view independent of the DBMS.
   b. Describes the information and functions of the organization without the use of any modelling language.
   c. Describes an information system in terms of the physical implementation of the database.
   d. Describes the relationships between objects in terms of the relational data model.

2 Logical independence is the property that ensures that the application programs are independent of...
   a. changes made to data that the application is using.
   b. the details of the conceptual representation of the data accessed by the application.
   c. the changes made to data that the application is not using.
   d. changes made to the physical implementation of the data

3 Select the **TRUE** statement regarding the end of a transaction
   a. User Commit - indicates the success of the transaction and changes are permanently saved.
   b. User Rollback - indicates that the user wants to save the changes made.
   c. System Commit - Indicates the success of the transaction, saving changes permanently.
   d. System Rollback - the transaction fails for some reason and the DBMS saves the operations executed.

4 Select the **TRUE** statement regarding the DB recovery from failures of system memory while using immediate update.
   a. Only data changes made by confirmed transactions are saved to disk.
   b. Changes made by confirmed transactions after the last checkpoint must be undone.
   c. Changes made for unconfirmed transactions after the last checkpoint must be undone.
   d. Changes made for unconfirmed transactions after the last checkpoint must be re-done.

5 Assuming deferred updates of transactions, if there is a system failure with loss of main memory, what actions should be taken?
   a. Redo only confirmed transactions after the last checkpoint.
   b. Undo unconfirmed transactions and redo confirmed transactions after the last checkpoint.
   c. Recover the last backup and redo confirmed transactions from the last checkpoint.
   d. Recover the last backup, undo unconfirmed transactions and redo confirmed transactions from the last checkpoint.

6 What SQL instructions should you use to define an external schema for a user?
   a. CREATE SCHEMA
   b. CREATE VIEW and CREATE DOMAIN
   c. CREATE VIEW and GRANT
   d. There are no instructions in SQL for this purpose, the external schemas are created from the applications that connect to the database.

7 Consider a DBMS whose transactions meet the property of isolation, which of the following situations **CANNOT** occur?

a. A transaction begins while others are in progress.
b. A transaction has read data that other unconfirmed transactions have read before.
c. A transaction blocks data that other unconfirmed transactions have read (and released) previously.
d. A T1 transaction reads an item of data that T2 has modified, and T2 has not been confirmed.

8 Select the **FALSE** statement

a. The atomicity property of a transaction means that either all the operations of the transaction are executed or none of them are executed.
b. The consistency property of a transaction guarantees that the transaction transforms the DB from one consistent state to another consistent state.
c. The isolation property of a transaction ensures that the execution of the transaction will not compromise the physical security of the DB.
d. The persistence of a transaction ensures that when the transaction is completed and confirmed, its changes will be recorded by the DB and will not be lost due to other transaction or system failures.

9 Consider the reconstruction of a database after a system failure with loss of **secondary** memory. If updates are executed in a deferred way, how are the transactions that appear cancelled in the journal file after the date of the last database backup processed?

a. The changes made by the transaction that are already recorded in the database will be undone.
b. These transactions are ignored in the recovery.
c. The DBMS will attempt to redo these transactions from the information in the journal file.
d. The information of the cancelled transactions is not available, since secondary memory has been lost and inconsistencies could appear in the recovery.

10 When a checkpoint is recorded in the daily file:

a. Transactions previous to the last failure are confirmed.
b. The updates produced by all the transactions confirmed in the log from the last checkpoint are recorded on disk.
c. The transactions in progress are cancelled and completely executed again after the checkpoint.
d. The changes already recorded of the transactions that are interrupted at that moment are undone.

ANSWERS

1	2	3	4	5	6	7	8	9	10
a	c	c	c	a	c	d	c	b	b

1.

A (a_0:t_a_0, a_1:t_a_1 a_0_x_2:t_a_0,…)

PK:{a_0}

FK:{a_0_x_2} → A(a_0)

B (b_0:t_b_0, b_1:t_b_1 , a_0:t_a_0, r:t_r …)

PK:{b_0, b_1}

FK:{a_0} → A

UNI:{a_0}

NNV{a_0, r}

D (b_0: t_b_0, b_1: t_b_1 , d_0:t_d_0 ,…)

PK:{b_0, b_1}

FK:{b_0,b_1} → B

NNV(d_0)

E (a_0:t_a_0, e_0:t_e_0, e_1:t_e_1 ,…)

PK:{a_0, e_0}

FK:{a_0} → A

UNI:{e_1}

F (f_0: t_f_0)

PK:{f_0}

G (g_0: t_c_0)

PK:{g_0,}

C (b_0: t_b_0, b_1: t_b_1 , c_0: t_c_0, …)

PK:{b_0, b_1}

FK:{b_0,b_1} → B

C1 (b_0: t_b_0, b_1: t_b_1 , c_1:t_c_1,,…)

PK:{b_0, b_1 , c_1}

FK:{b_0,b_1} → C

T (a_0:t_a_0, e_0:t_e_0, f_0: t_f_0 , t: t_t, g_0: t_g_0)

PK:{a_0, e_0}

FK:{a_0, e_0} → E

FK:{f_0} → F

FK:{g_0} → G

NNV:{f_0, t, g_0}

V (b_0:t_b_0, b_1:t_b_1, f_0: t_f_0)

PK:{b_0, b_1, f_0}

FK:{b_0, b_1} → D

FK:{f_0} → F

R.$I._{disjoint}$: There cannot be any pair of values (b_0, b_1) of B that are in (b_0, b_1) of C and in (b_0, b_1) of D

2.

R (A,B,G)

PK: {A,B}

FK: {B} →R2

NNV: {G}

RI1: Every (A,B) in R must be in R1

R1 (A,B,C)

PK: {A,B,C}

FK: {A,B} →R

R2 (B,D)

PK: {B}

FK: {D} →R21

NNV: {D}

RI2: Every B in R2 must be in R

R21 (D,E)

PK: {D}

FK: {E} →R22

NNV: {E}

RI3: Every D in R21 must be in R2

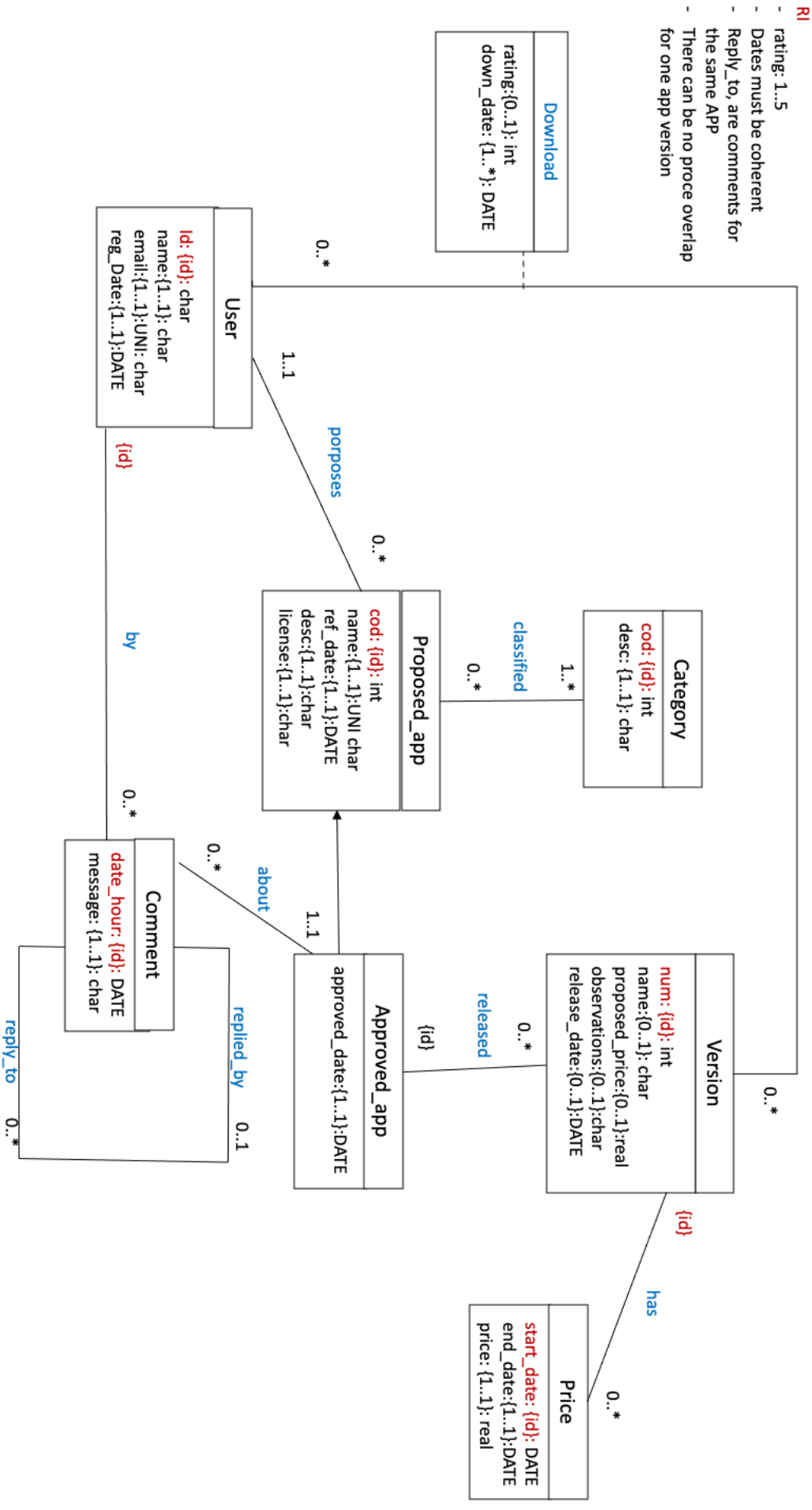
R22 (E,F)

PK: {E}

NNV: {F}

RI4: Every E in R22 must be in R21

3.


RI
- rating: 1..5
- Dates must be coherent
- Reply_to, are comments for the same APP
- There can be no proce overlap for one app version
Download
rating:{0..1}: int
down_date: {1..*}: DATE
User
Id: {id}: char
name:{1..1}: char
email:{1..1}:UNI: char
reg_Date:{1..1}:DATE
Category
cod: {id}: int
desc: {1..1}: char
Proposed_app
cod: {id}: int
name:{1..1}:UNI char
ref_date:{1..1}:DATE
desc:{1..1}:char
license:{1..1}:char
Version
num: {id}: int
name:{0..1}: char
proposed_price:{0..1}:real
observations:{0..1}:char
release_date:{0..1}:DATE
Approved_app
approved_date:{1..1}:DATE
Comment
date_hour: {id}: DATE
message: {1..1}: char
Price
start_date: {id}: DATE
end_date:{1..1}:DATE
price: {1..1}: real
0..*
1..1
porposes
0..*
1..*
classified
0..*
0..*
released
{id}
0..*
has
{id}
{id}
by
0..*
0..*
about
1..1
replied_by
0..1
reply_to
0..*