

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

05.567R15R06R19REEΘ€
05.567 15 06 19 EX

Enganxeu en aquest espai una etiqueta identificativa
amb el vostre codi personal
Examen

Aquest enunciat correspon també a les assignatures següents:

- 11.503 - Bases de dades

Fitxa tècnica de l'examen

- Comprova que el codi i el nom de l'assignatura corresponen a l'assignatura matriculada.
- Només has d'enganxar una etiqueta d'estudiant a l'espai corresponent d'aquest full.
- No es poden adjuntar fulls addicionals, ni realitzar l'examen en llapis o retolador gruixut.
- Temps total: **2 hores** Valor de cada pregunta: **indicat a cada exercici**
- En cas que els estudiants puguin consultar algun material durant l'examen, quins són?
no es pot fer servir material

En cas de poder fer servir calculadora, de quin tipus? **CAP**

- Si hi hagi preguntes tipus test: Descompten les respostes errònies? **NO** Quant?
- Indicacions específiques per a la realització d'aquest examen:

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

Enunciats

EXERCICI 1 (3 punts)

Volem disposar d'una base de dades per enregistrar informació sobre pel·lícules, actors i directors. A continuació, es descriu cadascuna de les taules.

PERSON (PERSONA)

La taula *PERSON* conté les dades generals sobre les persones que apareixen a la BD. En concret, per a cada persona s'enregistra un nombre identificador (*id*) que és clau primària, el nom (*name*), la data de naixement (*birth*), la data de defunció (*death*), el gènere (*gender*), la nacionalitat (*nationality*) i l'adreça (*address*). Només els atributs data de defunció (*death*) i adreça (*address*) poden prendre valors nuls.

DIRECTOR (DIRECTOR)

La taula conté informació sobre els directors. En concret, per cada director, es guarda l'identificador (*id*) que és clau primària i clau forana de *PERSON*. L'any de debut (*debut_year*) i el nombre de premis que ha rebut durant la seva carrera (*num_awards*). Només l'atribut any de debut (*debut_year*) pot ser nul.

ACTOR (ACTOR)

Guarda informació sobre els actors. Per cada actor es guarda l'identificador (*id*) que és clau primària i clau forana de *PERSON* i l'any de debut (*debut_year*), que pot ser nul.

MOVIE (PEL·LÍCULA)

Informació sobre les pel·lícules. Per a cada pel·lícula s'enregistra el seu identificador (*id*) que és clau primària, el títol (*title*), el gènere (*genre*), l'any d'estrena (*year*), el país de producció (*country*), la duració en minuts (*duration*), el pressupost en euros (*budget*), el ingressos generats en euros (*incomes*), una referència a la pel·lícula anterior si la pel·lícula pertany a una saga (*id_movie_prev*) i l'identificador del director (*id_director*) que és clau forana a *DIRECTOR*.

Només l'atribut que fa referència a la pel·lícula anterior pot prendre valors nuls. L'atribut *duration* ha de ser estrictament positiu i els atributs *incomes* i *budget* han de ser iguals o majors a zero.

L'atribut gènere només pot prendre els valors {'Action', 'Adventure', 'Comedy', 'Crime', 'Drama', 'Historical', 'Horror', 'Musical', 'Science-fiction', 'War', 'Western'}.

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

PARTICIPATION (PARTICIPACIÓ)

Informació sobre les participacions dels actors a les pel·lícules. S'emmagatzemen el valors $\{id_actor, id_movie\}$ que són clau primària, on $\{id_actor\}$ és clau forana de *ACTOR* i $\{id_movie\}$ és clau forana de *MOVIE*. També es guarden el nom del personatge interpretat (*character*) i el rol (*role*) que només pot prendre els valors $\{'Leading', 'Supporting', 'Narrator', 'Cameo', 'Unclassified'\}$.

Cap atribut pot prendre valors nuls. Si s'esborra una pel·lícula de la BD, caldrà esborrar totes les participacions d'actors a la pel·lícula.

(1 punt) ES DEMANA, utilitzant expressions d'àlgebra relacional:

1) Obtenir els actors nascuts després de 1950 que han estat també directors abans del 2018. En concret es demana l'identificador, el nom i l'any de naixement de l'actor.

```

R1:= ACTOR * PERSON
R2:= R1 {birth > 1950}
R3:= R2[id]
R4:= DIRECTOR {debut_year < 2018}
R5:= R4 [id]
R6:= R3 ∩ R5
R7:= R6 * PERSON
R8:= R7[id, name, birth]
  
```

A R3 tenim els identificadors dels actors nascuts després de 1950. A R5 tenim els identificadors dels directors que han debutat com a tal abans de l'any 2018. A R6 tenim els identificadors dels actors que també han estat directors en les condicions que es demanaven. A R7 recuperem les dades d'aquestes persones, i a R8 tenim els atributs demanats com a resultat de la consulta.

(1 punt) ES DEMANA, utilitzant expressions d'SQL:

2) Trobar la pel·lícula dirigida per un director de nacionalitat no americana que ha obtingut els ingressos màxims. En concret es vol l'identificador de la pel·lícula, el títol de la pel·lícula i els seus ingressos, així com el nom i la nacionalitat del director, si ni ha més d'una, ordenades per nacionalitat.

```

SELECT m.id, m.title, m.incomes, p.name, p.nationality
FROM MOVIE M INNER JOIN PERSON P ON (M.id_director = P.id)
WHERE M.incomes = SELECT ( MAX(incomes)
                      FROM MOVIE M1 INNER JOIN PERSON P1 ON ( M1.id_director = P1.id)
                      WHERE P1.nationality <> 'American')
      AND P.nationality <> 'American')
ORDER BY nationality;
  
```

(1 punt) ES DEMANA, donada la seqüència següent d'expressions d'àlgebra relacional:

```
R1:= PARTICIPATION (role= 'Leading')
```

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

R2:= R1[id_actor]
R3:= PARTICIPATION (role = 'Narrator')
R4:= R3[id_actor]
R5:= R2 - R4
R6:= ACTOR * PERSON
R7:= R6 [id = id_actor] R5
R9:= R8[id, name, debut_year]

3.1) **(0.5 punts)** Explicar en llenguatge natural què fa la seqüència d'operacions prèvies (en definitiva, es tracta de donar l'enunciat de la consulta que es planteja).

Obtenir els actors que han participat en alguna pel·lícula com a actor/actriu principal (*leading*) i no han participat mai en cap pel·lícula com a narrador (*narrator*). Concretament volem l'identificador, el nom, i l'any de debut d'aquests actors.

3.2) **(0.5 punts)** Proposar una consulta equivalent en llenguatge SQL.

```

SELECT P.id, P.name, A.debut_year
FROM PERSON P INNER JOIN ACTOR A ON P.Id = A.Id INNER JOIN PARTICIPATION PA ON
A.id = PA.id_actor
WHERE PA.role = 'Leading' AND
      A.id NOT IN (SELECT id_actor FROM PARTICIPATION WHERE role = 'Narrator');

```

EXERCICI 2 (2.5 punts)

Sobre la base de dades de pel·lícules del primer exercici, hem afegit la columna nombre de participacions com a actor principal (*num_leading*) a la taula *ACTOR*. Aquesta columna indica el nombre total de vegades que l'actor ha participat com a actor principal en alguna pel·lícula.

- Volem mantenir el càlcul del nombre de participacions com a actor principal (*num_leading*) de l'actor actualitzat. En concret, volem que aquesta columna sempre reflecteixi el nombre total de vegades que l'actor ha tingut el rol de '*Leading*' en alguna participació. Podem suposar que els usuaris o programes mai actualitzaran directament la columna *num_leading* de la taula *ACTOR*, i que en el moment d'inserir un nou actor, el valor de la columna *num_leading* serà zero.

Digueu quins elements d'SQL faríeu servir per implementar en PostgreSQL les regles anteriors.

En cas que trieu disparadors, cal indicar l'operació (*INSERT*, *DELETE*, *UPDATE*) que activaria l'execució de cada disparador i les taules implicades. En cas d'operacions d'*UPDATE* cal especificar les columnes rellevants.

Adicionalment, indiqueu el tipus de disparador (*FOR EACH ROW/FOR EACH STATEMENT* i si ha de ser *BEFORE/AFTER*), així com una breu descripció de les comprovacions que caldria realitzar.

Per a la regla de negoci cal definir disparadors sobre les taules i esdeveniments següents:

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

	<i>INSERT</i>	<i>UPDATE</i>	<i>DELETE</i>
<i>PARTICIPATION</i>	<p>S'afegirà una nova participació. El valor de la columna <i>num_leading</i> de l'actor relacionat s'incrementa en una unitat si el rol de la participació és igual a 'Leading'.</p> <p><i>FOR EACH ROW</i>, cal accedir a valors de la fila a inserir mitjançant <i>NEW</i>.</p> <p><i>AFTER</i> així garantim que no s'ha violat cap regla de negoci.</p>	<p>En cas d'<i>UPDATE</i> hem de tenir en compte la modificació de les columnes següents:</p> <p><i>id_actor</i>: si canviem aquesta columna, hem de tornar a calcular el valor de la columna <i>num_leading</i> per a l'antic actor i per al nou sempre que l'<i>UPDATE</i> es faci sobre una participació amb el valor <i>rol</i> igual a 'Leading'. A l'actor antic es resta una unitat i al nou es suma una unitat.</p> <p><i>role</i>: si canviem aquesta columna, hem de tornar a calcular el valor de la columna <i>num_leading</i> quan l'antic valor o el nou sigui 'Leading'. Se suma una unitat si el nou rol és 'Leading' (i no ho era l'antic) i es resta si el nou valor no és 'Leading' (i sí que ho era l'antic).</p> <p><i>FOR EACH ROW</i>, ja que les sentències d'<i>UPDATE</i> poden actuar sobre un conjunt de files (accedirem als valors de la fila mitjançant <i>NEW</i> i <i>OLD</i>).</p> <p><i>AFTER</i> així garantim que no s'ha violat cap regla de negoci.</p>	<p>En cas d'esborrar una fila de la taula, s'haurà de tornar a calcular el valor de la columna <i>num_leading</i> per a l'actor relacionat, sempre que la participació fos del tipus 'Leading'.</p> <p><i>FOR EACH ROW</i>, cal accedir a valors de la fila a esborrar mitjançant <i>OLD</i> per restar una unitat a la columna <i>num_leading</i> de l'actor relacionat.</p> <p><i>AFTER</i> així garantim que no se ha violat cap regla de negoci.</p>

EXERCICI 3 (2.5 punts)

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

Sigui un SGBD **sense cap mecanisme de control de concurrència**, i suposem que es produeix l'horari que mostrem a continuació (on R=lectura, RU=lectura amb intenció d'actualització, W=escriptura; les accions s'han numerat per facilitar la seva referència):

Acc#	T1	T2
10		RU(A)
20		W(A)
30		R(E)
40		RU(B)
50	RU(B)	
60		W(B)
70	W(B)	
80		RU(E)
90		W(E)
100	RU(C)	
110	W(C)	
120		R(E)
130		R(C)
140	R(A)	
150		COMMIT
160	COMMIT	
170		

Responen, justificant **breument** la vostra resposta, les qüestions següents:

- 1) **(1 punt)** Indiqueu la interferència o interferències que s'hi produeixen: tipus, grànuls i transaccions implicades, així com el nivell mínim d'aïllament d'SQL amb el qual haurien de treballar les transaccions per tal d'evitar totes les interferències trobades simultàniament.
- 2) **(0.5 punts)** L'horari proposat, és recuperable? Quins horaris serials donarien els mateixos resultats que l'horari proposat?
- 3) **(1 punt)** Apliqueu (indicant com queda l'horari) un mecanisme de control de concurrència basat en reserves S, X i on les transaccions treballen amb un nivell d'aïllament REPEATABLE READ. Quins horaris serials donarien els mateixos resultats que l'horari resultant?

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

1) Les interferències que es produeixen entre T1 i T2 són:

Actualització perduda: grànul B

Anàlisi inconsistent: grànul A i C

Acc#	T1	T2
10		RU(A)
20		W(A)
30		R(E)
40		RU(B)
50	RU(B)	
60		W(B)
70	W(B)	
80		RU(E)
90		W(E)
100	RU(C)	
110	W(C)	
120		R(E)
130		RU(C)
140	RU(A)	
150		COMMIT
160	COMMIT	
170		

El nivell d'aïllament d'SQL per evitar la interferència d'actualització perduda és READ UNCOMMITTED i la de l'anàlisi inconsistent és REPEATABLE READ. Per evitar simultàniament les dues interferències ens hem de quedar amb el nivell més restrictiu, és a dir REPEATABLE READ.

2) L'horari no és recuperable (per exemple, T2 llegeix C a l'instant 130, aquest grànul ha estat modificat per T1 a l'instant 110, i T2 acaba la seva execució abans que T1). No hi ha cap horari serial equivalent ja que l'horari proposat no és serialable (hem trobat 2 interferències a l'apartat previ).

3) L'horari queda tal i com s'indica a continuació:

Acc#	T1	T2
10		LOCK(A,X)
20		RU(A)
30		W(A)
40		LOCK(E,S)
50		R(E)
60		LOCK(B,X)
70		RU(B)

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

80	LOCK(B,X)	
90		W(B)
100		LOCK(E,X)
110		RU(E)
120		W(E)
130		R(E)
140		LOCK(C,S)
150		R(C)
160		COMMIT (UNLOCK(A), UNLOCK(E), UNLOCK(B), UNLOCK(C))
170	RU(B)	
180	W(B)	
190	LOCK(C,X)	
200	RU(C)	
210	W(C)	
220	LOCK(A,S)	
230	R(A)	
240	COMMIT (UNLOCK(B), UNLOCK (C), UNLOCK(A))	

T1 bloqueja la seva execució en l'instant 90 donat que no pot adquirir la reserva que necessita i que està en possessió de T2 fins el seu acabament. L'horari resultant dona els mateixos resultats que l'horari serial T2;T1.

EXERCICI 4 (2 punts)

1) (1 punt) Donat el codi següent:

```
private void run() {
    DBAccessor dbaccessor = new DBAccessor();
    dbaccessor.init();
    Connection conn;
    try {
        conn = dbaccessor.getConnection();
        conn.setAutoCommit(false);

        if (conn != null) {
            Statement stmt = conn.createStatement();
```


Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

```

        ResultSet rs = stmt.executeQuery("SELECT * FROM
PERSON");

        while (rs.next()) {
            String name = rs.getString(1);
            System.out.println("Nom persona :" + name);
        }
    }
}
catch(Exception e) {
    System.err.println("Error ocurred");
}
}

```

Explica quins errors veus i proposa quins canvis faries al codi per solucionar-los. **Extensió màxima mitja pàgina.**

El problema amb el codi anterior és que els recursos com la connexió a la base de dades, el Statement o el ResultSet no es tanquen ni en el cas que no hi ha cap error ni quan hi ha una excepció. La manera de solucionar aquest problema és posant un bloc finally on es tanquin tots els recursos. El bloc finally garanteix que s'executa sempre, fins i tot quan hi ha excepcions.

```

        catch(Exception e) {
            System.err.println("Error ocurred");
        } finally {
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
            try {
                stmt.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
            try {
                rs.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}

```

2) (1 punt) Explica els avantatges de les bases de dades analítiques respecte del model relacional. **Extensió màxima mitja pàgina.**

Els avantatges principals d'un magatzem de dades respecte d'una base de dades relacional són els següents:

1) Permeten representar informació que pot ser útil per a la presa de decisions, en comptes de ser-ho per a l'operativa diària en l'organització.

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	15/06/2019	15:30

2) Suporten la integració de dades provinents de fonts diverses, cosa que permet l'obtenció d'un model de dades comú dissenyat per a treure rendiment de grans volums de dades històriques.

3) Són pensades per a l'anàlisi de dades amb eines OLAP, no per a actualitzacions en línia.

4) Faciliten la integració de magatzems de dades en entorns de treball d'usuaris que tenen un perfil analític o gerencial.