

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	08/06/2019	12:00

05.567 08 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	08/06/2019	12:00

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	08/06/2019	12:00

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 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.

```

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

```

A R2 tenim els identificadors dels actors que han participat en alguna pel·lícula com a protagonista (*Leading*). A R4 tenim els identificadors dels actors que han participat en alguna pel·lícula com a narrador (*Narrator*). A R5 tenim els identificadors dels actors que han participat com a protagonista en alguna pel·lícula però mai com a narrador en cap pel·lícula. A R6 i R7 recuperem les dades d'aquests actors, i a R8 tenim els atributs demanats com a resultat de la consulta.

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

2) 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, ordenats per any de naixement.

```

SELECT P.id, P.name, P.birth
FROM PERSON P NATURAL JOIN DIRECTOR D NATURAL JOIN ACTOR A
WHERE P.birth > 1950 AND D.debut_year < 2018
ORDER BY P.birth;

```

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

```

R1:= PERSON (nationality <> 'American');
R2:= R1 * DIRECTOR

```

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	08/06/2019	12:00

```

R3(id_director):= R2(id)
R4:= MOVIE * R3
R5:= R4[id, incomes]
R6(id1,incomes1):= R5
R7:=R5[incomes<incomes1]R6
R8:= R7[id,incomes]
R9:= R5 - R8
R10:= R9 * R4
R11:= R10 [id, title, incomes, name, nationality]

```

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).

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.

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

```

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';

```

EXERCICI 2 (2.5 punts)

Sobre la base de dades de pel·lícules del primer exercici, hem afegit la columna nombre total de pel·lícules amb beneficis de l'actor (*num_profit_movies*) a la taula *ACTOR*. Aquesta columna indica el nombre total de pel·lícules en les quals ha participat l'actor tals que els seus ingressos (*incomes*) han estat superiors al pressupost (*budget*).

- Volem mantenir el càlcul del nombre total de pel·lícules amb beneficis (*num_profit_movies*) de l'actor actualitzat. En concret, volem que aquesta columna sempre reflecteixi el nombre total de les pel·lícules on ha participat l'actor i han tingut beneficis. Podem suposar que els usuaris o programes mai no actualitzaran directament la columna *num_profit_movies* de la taula *ACTOR*, i que en el moment d'inserir un nou actor, el valor de la columna *num_profit_movies* 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.

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	08/06/2019	12:00

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

	INSERT	UPDATE	DELETE
MOVIE		<p>En cas d'<i>UPDATE</i> hem de tenir en compte la modificació de les columnes següents:</p> <p><i>incomes</i> o <i>budget</i>: Si canviem una d'aquestes columnes, hem de tornar a calcular el valor de la columna <i>num_profit_movies</i> sempre que la diferència entre els valors antics i els valors nous de <i>incomes</i> i <i>budget</i> sigui diferent (abans donava beneficis i ara no, o a la inversa). Segons el cas, sumarem o restarem una unitat a <i>num_profit_movies</i> a tots els actors que han participat a la pel·lícula.</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>	
PARTICIPATION	<p>S'afegirà una nova participació. El valor de la columna <i>num_profit_movies</i> de l'actor relacionat s'incrementa en una unitat si els <i>incomes</i> de la pel·lícula a què fa referència la participació són majors</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_profit_movies</i> per a</p>	<p>En cas d'esborrar una fila de la taula, s'haurà de tornar a calcular el valor de la columna <i>num_profit_movies</i> per a l'actor relacionat, sempre que la pel·lícula tingués beneficis.</p> <p><i>FOR EACH ROW</i>, cal accedir a valors de la fila a esborrar mitjançant <i>OLD</i> per restar una</p>

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	08/06/2019	12:00

que el <i>budget</i> .	l'antic actor i per al nou, si unitat a columna
<i>FOR EACH ROW</i> , cal la participació fa	<i>num_profit_movies</i> de l'actor
accedir a valors de la referència a una pel·lícula relacionat.	
fila a inserir mitjançant amb beneficis.	
<i>NEW</i> .	<i>AFTER</i> així garantim que no
<i>AFTER</i> així garantim s'ha violat cap regla de negoci.	s'ha violat cap regla de negoci.
	<i>id_movie</i> : si canviem
	aquesta columna, hem
	de tornar a calcular el
	valor de la columna
	<i>num_profit_movies</i> per a
	l'actor amb les dades de
	l'antiga pel·lícula i la
	nova, si la participació fa
	referència a una pel·lícula
	amb beneficis.
	<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>).
	<i>AFTER</i> així garantim que
	no s'ha violat cap regla
	de negoci.

EXERCICI 3 (2.5 punts)

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(F)
20		W(F)
30	R(F)	
40	RU(A)	
50	W(A)	
60		RU(F)
70		W(F)
80		R(C)
90	RU(C)	
100	W(C)	
110		RU(E)
120		W(E)
130		R(C)

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	08/06/2019	12:00

140	COMMIT	
150		COMMIT
160		

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 de 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 READ COMMITTED. Quins horaris serials donarien els mateixos resultats que l'horari resultant?

- 1) Les interferències que es produeixen són:

Lectura no confirmada entre T1 i T2 sobre el grànul F

Lectura no repetible entre T1 i T2 sobre el grànul C.

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

El nivell d'aïllament mínim per evitar la lectura no confirmada és READ COMMITTED, i per evitar la lectura no repetible 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, T1 llegeix F a l'instant 30, aquest grànul ha estat modificat per T2 a l'instant 20, i T1 acaba la seva execució abans que T2). No hi ha cap horari serial equivalent ja que l'horari proposat no és serialable (hem trobat 2 interferències a l'apartat 1).

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	08/06/2019	12:00

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

Acc#	T1	T2
10		LOCK(F,X)
20		RU(F)
30		W(F)
40	LOCK(F,S)	
50		RU(F)
60		W(F)
70		LOCK(C,S)
80		R(C)
90		UNLOCK(C)
100		LOCK(E,X)
110		RU(E)
120		W(E)
130		LOCK(C,S)
140		R(C)
150		UNLOCK(C)
160		COMMIT (UNLOCK(F), UNLOCK(E))
170	R(F)	
180	UNLOCK(F)	
190	LOCK(A,X)	
200	RU(A)	
210	W(A)	
220	LOCK(C,X)	
230	RU(C)	
240	W(C)	
250	COMMIT (UNLOCK(A), UNLOCK(C))	
260		
270		
280		
290		

T1 bloqueja la seva execució en l'instant 50 perquè no pot adquirir una reserva sobre F que està en possessió de T2 fins al seu acabament.

Tot i que READ COMMITTED no és un nivell suficient per evitar la interferència de lectura no repetible, veiem que en aquest cas, i degut al bloqueig de T1, a l'horari resultant s'eviten totes dues interferències.

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	08/06/2019	12:00

L'horari resultat dona els mateixos resultats que l'horari serial T2;T1.

EXERCICI 4 (2 punts)

1) (1 punt) Donat el següent codi:

```
private void run(String gender) {
    DBAccessor dbaccessor = new DBAccessor();
    dbaccessor.init();
    Connection conn = dbaccessor.getConnection();
    ResultSet rs = null;
    PreparedStatement stmt = null;
    try {
        conn.setAutoCommit(false);

        if (conn != null) {
            stmt = conn.prepareStatement("SELECT * FROM PERSON
WHERE gender = " + gender );

            rs = stmt.executeQuery();
            while(rs.next()) {
                String name = rs.getString(1);
                System.out.println("Nom persona :" + name);
            }
        }
    }
    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();
        }
    }
}
```

Examen 2018/19-2

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	08/06/2019	12:00

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 la utilització incorrecta del PreparedStatement, ja que estem composant directament la Query ("SELECT * FROM PERSON WHERE gender = " + gender) i això pot causar un problema de seguretat, concretament seríem vulnerables a atacs de SQL injection. Per a solucionar l'error utilitzaríem la funció setString de PreparedStatement.

```
stmt = conn.prepareStatement("SELECT * FROM PERSON WHERE gender = ?" );  
stmt.setString(1, gender);
```

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

Els avantatges principals de les bases de dades deductives respecte de les bases de dades relacionals són els següents:

- 1) Possibiliten la inferència d'informació a partir de les dades emmagatzemades i les regles de deducció que els són aplicables. El model relacional no disposa de mecanismes d'inferència automàtics; el que més s'hi assembla són els disparadors, però cal implementar-los.
- 2) Suporten objectes complexos i conjunts definits a partir de la composició de predicats de la lògica, de manera que són recuperables mitjançant consultes recursives i algorismes eficients per a aquest tipus de dades. El model relacional només admet estructures de dades simples que permeten la uniformitat en el tractament de la informació mitjançant un llenguatge bàsicament declaratiu, que no resulta idoni per a la recursivitat.
- 3) Tenen capacitat per a expressar consultes mitjançant regles lògiques, fet que permet automatitzar les especificacions d'aquest tipus de consultes. El model relacional disposa de vistes per a deduir nova informació, però la seva especificació i la implementació es duen a terme separatament.