

## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

05.567 09 01 19 EX  
05.567 09 01 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:

### Enunciats

## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

### 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 el 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 referencia 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 referencia a la pel·lícula anterior pot prendre valors nuls. L'atribut *duration* ha de ser positiu estricte 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'}.

#### PARTICIPATION (PARTICIPACIÓ)

Informació sobre les participacions dels actors a les pel·lícules. S'emmagatzemen el valors {*id\_actor*, *id\_movie*} que son clau primària, on {*id\_actor*} és clau forana de *ACTOR* i {*id\_movie*} és

## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

clau forana de *MOVIE*. 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) Obténir els actors que han participat en alguna pel·lícula de ciència ficció (*Science-fiction*), però mai en pel·lícules bèl·liques (*War*). Concretament volem l'identificador, el nom, i l'any de debut d'aquests actors.

R1:= MOVIE(genre= 'Science-fiction')

R2:= R1[id=id\_movie]PARTICIPATION

R3:= R2[id\_actor]

R4:= MOVIE(genre= 'War')

R5:= R4[id=id\_movie]PARTICIPATION

R6:= R5[id\_actor]

R7:= R3-R6

R8:= R7[id\_actor=id]ACTOR

R9:= R8\*PERSON

R10:= R9[id, name, debut\_year]

A R3 tenim l'identificador dels actors que han participat a pel·lícules de ciència ficció. Per una altra banda a R6 tenim l'identificador dels actors que han participat a pel·lícules bèl·liques. A R7 tenim l'identificador dels actors que han participat a pel·lícules de ciència ficció, però que mai han participat a pel·lícules bèl·liques. A R8 recuperem les dades d'aquests actors, i a R9 les dades personals dels actors. Finalment a R10 tenim els atributs demanats com a resultat de la consulta.

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

2) Trobar les pel·lícules que a l'any 2018 han tingut uns ingressos superiors a la mitjana d'ingressos de les pel·lícules del mateix any. En concret es vol l'identificador de la pel·lícula, el nom de la pel·lícula, l'identificador del director, el nom del director i l'any de debut del director.

## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

```
SELECT m.id, m.title, m.id_director, p.name, d.debut_year
FROM MOVIE m JOIN DIRECTOR d ON m.id_director=d.id
      JOIN PERSON p ON d.id=p.id
WHERE m.year=2018 AND
      m.incomes > (SELECT AVG(m1.incomes)
                    FROM MOVIE m1
                    WHERE m1.year=2018);
```

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

R1:= MOVIE(year=2017)

R2:= R1[id]

R3:= R2[id=id\_movie]PARTICIPATION

R4:= R3[id\_actor, id\_movie]

R5(id\_actor1, id\_movie1):= R4

R6:= R5[id\_actor1=id\_actor, id\_movie1<>id\_movie]R4

R7:= R6[id\_actor=id]ACTOR

R8:= R7\*PERSON

**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 més d'una pel·lícula a l'any 2017. 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 pe.id, pe.name, a.debut_year
FROM MOVIE m JOIN PARTICIPATION p ON m.id=p.id_movie
      JOIN ACTOR a ON p.id_actor=a.id
      JOIN PERSON pe ON pe.id=a.id
WHERE m.year=2017
GROUP BY pe.id, pe.name, a.debut_year
HAVING COUNT(*)>1;
```

## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

### EXERCICI 2 (2.5 punts)

Sobre la base de dades de pel·lícules del primer exercici, hem afegit la columna ingressos totals de l'actor (*total\_incomes*) a la taula *ACTOR*. Aquesta columna indica el total d'ingressos aconseguits per les pel·lícules en les que ha participat l'actor.

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

	<i>INSERT</i>	<i>UPDATE</i>	<i>DELETE</i>
<i>PARTICIPATION</i>	<p>S'afegirà una nova participació (el valor de la columna <i>total_incomes</i> s'incrementa amb el valor <i>incomes</i> de la pel·lícula a l'actor relacionat).</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>total_incomes</i> per a l'antic actor i per al nou.</p> <p><i>id_movie</i>: Si canviem aquesta columna, hem de tornar a calcular el valor de la columna <i>total_incomes</i> per l'actor relacionat.</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</p>	<p>En cas d'esborrar una fila de la taula, s'haurà de tornar a calcular el valor de la columna <i>total_incomes</i> per l'actor relacionat.</p> <p><i>FOR EACH ROW</i> (cal accedir a valors de la fila a esborrar mitjançant <i>OLD</i> per restar el valor de <i>incomes</i> de la pel·lícula a <i>total_incomes</i> de l'actor relacionat).</p> <p><i>AFTER</i> així garantim que no s'ha violat cap regla de negoci.</p>

## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

		la fila mitjançant <i>NEW</i> i <i>OLD</i> ).	
		<i>AFTER</i> així garantim que no s'ha violat cap regla de negoci.	
<i>MOVIE</i>		En cas d' <i>UPDATE</i> hem de tenir en compte la modificació de les columnes següents:  <i>incomes</i> : Si es canvia els ingressos de la pel·lícula, s'ha de recalculer <i>total_incomes</i> (restant els antics i sumant els nous valors).	

### 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	T3
10	R(E)		
20			R(B)
30		RU(C)	
40		W(C)	
50		R(E)	
60		RU(B)	
70		W(B)	
80			R(C)
90	RU(A)		
100	W(A)		
110			R(E)
120		R(A)	
130	R(D)		
140	RU(A)		

## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

150	W(A)		
160		COMMIT	
170			COMMIT
180	COMMIT		

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 que haurien de treballar les transaccions per tal d'evitar simultàniament totes les interferències trobades.

Acc#	T1	T2	T3
10	R(E)		
20			R(E)
30		RU(C)	
40		W(C)	
50		R(E)	
60		RU(B)	
70		W(B)	
80			R(C)
90	RU(A)		
100	W(A)		
110			R(E)
120		R(A)	
130	R(D)		
140	RU(A)		
150	W(A)		
160		COMMIT	
170			COMMIT
180	COMMIT		

Anàlisi inconsistent entre T2 i T3 pels grànuls B i C.  
Lectura no confirmada entre T1 i T2 pel grànul A.

El nivell d'aïllament mínim d'SQL per evitar la interferència d'anàlisi inconsistent és REPEATABLE READ i per evitar la interferència de lectura no confirmada és READ COMMITTED.

## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

Per tant, per evitar totes dues interferències simultàniament ens hem de quedar amb el nivell més restrictiu, és a dir, REPEATABLE READ.

2) **(0.5 punts)** L'horari proposat, és recuperable? Quins horaris serials donarien els mateixos resultats que l'horari proposat?

L'horari proposat no és recuperable, ja que, per exemple, T2 a l'acció 120 llegeix el grànul A, prèviament escrit per T1 a l'acció 100, i T2 acaba la seva execució confirmant abans que T1.

No hi ha cap horari serial equivalent ja que l'horari proposat no és serialable degut a que hi ha dues interferències.

3) **(1 punt)** Apliqueu (indicant com quedaria 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?

Acc#	T1	T2	T3
10	LOCK(E,S)		
20	R(E)		
30	UNLOCK(E)		
40			LOCK(B,S)
50			R(B)
60			UNLOCK(B)
70		LOCK(C,X)	
80		RU(C)	
90		W(C)	
100		LOCK(E,S)	
110		R(E)	
120		UNLOCK(E)	
130		LOCK(B,X)	
140		RU(B)	
150		W(B)	



## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

160			LOCK(C,S)
170	LOCK(A,X)		
180	RU(A)		
190	W(A)		
200		LOCK(A,S)	
210	LOCK(D,S)		
220	R(D)		
230	UNLOCK(D)		
240	RU(A)		
250	W(A)		
260	COMMIT UNLOCK(A)		
270		R(A)	
280		UNLOCK(A)	
290		COMMIT UNLOCK(C) UNLOCK(B)	
300			R(C)
310			UNLOCK(C)
320			LOCK(E,S)
330			R(E)
340			UNLOCK(E)
350			COMMIT

Tot i que T3 bloqueja la seva execució a l'acció 160 per adquirir la reserva de lectura sobre el grànul C, i T2 bloqueja la seva execució a l'acció 200 per adquirir la reserva de lectura sobre el grànul A, veiem que amb el nivell READ COMMITTED evitem la interferència de lectura no confirmada però seguim tenint la interferència d'anàlisi inconsistent. Per tant, no hi ha cap horari serial equivalent. L'estona que cada transacció està bloquejada es marca en gris a la solució.

## Examen 2018/19-1

Assignatura	Codi	Data	Hora inici
Ús de bases de dades	05.567	09/01/2019	12:00

### EXERCICI 4 (2 punts)

1) **(1 punt)** Explica què fan les següents línies de codi usat per resoldre el tercer lliurament de la pràctica. **Extensió màxima mitja pàgina.**

```
StringBuffer sbUrl = new StringBuffer();
sbUrl.append("jdbc:postgresql:");
if (host != null && !host.equals("")) {
    sbUrl.append("//").append(host);
    if (port != null && !port.equals("")) {
        sbUrl.append(":").append(port);
    }
}
sbUrl.append("/").append(dbname);
url = sbUrl.toString();
```

Preparen la connexió a la base de dades, construint la `url` on s'ha d'adreçar la connexió.

2) **(1 punt)** Què és una base de dades analítica (*data warehouse*) i quina és la seva utilitat principal? En el context d'aquest tipus de base de dades, què és un model multidimensional? **Extensió màxima mitja pàgina.**

Un *data warehouse* és un tipus de base de dades que permet emmagatzemar i recuperar informació històrica que prové de diferents fonts de dades internes (en general, bases de dades operacionals, que són les que hem estudiat a l'assignatura) i externes amb l'objectiu d'oferir suport a la presa de decisions.

El model multidimensional és la forma en què les dades acostumen a estar organitzades en el *data warehouse* per tal que aquestes puguin ser explotades convenientment. Les dades s'organitzen en forma de fets, dimensions i mesures que posteriorment s'utilitzen per a l'anàlisi de la informació. Els fets representen l'activitat que es vol analitzar. Les dimensions caracteritzen l'activitat, és dir, materialitzen els diversos punts de vista des dels quals es pot analitzar l'activitat. Les mesures representen la informació rellevant sobre els fets.