

Preguntes test
Disseny de Bases de Dades (DBD)

July 18, 2025

Grau d'Enginyeria Informàtica
Universitat Politècnica de Catalunya (BarcelonaTech), Barcelona

Contents

0	Introduction	5
0.1	Basic background	5
0.2	Kinds of systems	6
0.3	Design steps	7
0.4	Kinds of models	8
1	Relational Translation - Difficulties, Criteria and Tools	9
1.1	Design Steps	9
1.2	Interpretation	9
1.3	Semantic Relativism	9
1.4	Null Values	10
1.5	Generalization/Specialization	12
1.6	Surrogates	13
2	Relational Translation - Relationships	15
2.1	Constraints	15
2.2	Binary Associations	16
2.3	Reflexive Associations	17
2.4	N-ary Associations	17
2.5	Compound Aggregations	18
2.6	Association Classes	18
2.7	Multivalued Attributes	19
2.8	Example of Translation	19
3	Normalization	20
3.1	Anomalies	20
3.2	Functional Dependencies	20
3.3	Normal Forms	21
3.4	Automatic Normalization	22
3.5	Examining Normalization	23
3.6	4NF	24
4	Data Warehousing and OLAP	25
4.1	Operational and Decisional Comparison	25
4.2	Data Warehouse Definition	26
4.3	Data Warehousing Architectures	27
4.4	OLAP Definition	28
4.5	OLAP Operations	28
4.6	Conceptual Multidimensional Schema	29
4.7	Logical Multidimensional Schema	29
4.8	SQL Extension	30

5	NOSQL	32
5.1	Motivation	32
5.2	Schema Definition	32
5.3	Storing Arrays	33
5.4	Semi-structured Database Model	34
5.5	JSON Data Type	35
6	Views	36
6.1	View Definition and Problems	36
6.2	View Expansion	38
6.3	Update Through Views	38
6.4	View Updating	39
6.5	Query Rewriting	40
6.6	Materialized View Selection	40
6.7	Example of Materialize View Selection	41
7	Physical Design	42
7.1	Tasks, Criteria and Difficulties	42
7.2	Catalog	42
7.3	Access Structures	43
7.4	Size of Structures	45
7.5	Access Costs	45
7.6	Index Choice	46
8	Query Optimization Phases: Semantic, Syntactic and Physical	47
8.1	Optimization Phases	47
8.2	Semantic Optimization	47
8.3	Syntactic Optimization	48
8.4	Equivalence Rules	49
8.5	Physical Optimization	50
9	Query Optimization Costs: Selection, Sorting and Projection	52
9.1	Intermediate Results	52
9.2	Selection	54
9.3	Sorting	55
9.4	Projection	56
10	Query Optimization Costs: Join	57
10.1	Clustered Structure	57
10.2	Nested Loops	57
10.3	Hash-Join	58
10.4	Sort-Match	59
11	Parametrization and Tuning	60
11.1	Tablespaces	60
11.2	Parameters	61
11.3	Bitmap Indexes	61
11.4	Tuning	63

11.5 Example of Index Selection	64
12 Transactions	65
12.1 Isolation	65
12.2 Multi-version Concurrency Control	66
12.3 Atomicity and Durability	67
12.4 Transaction Chopping	69
12.5 Alternative Transaction Manager Architectures	70
A Respostes	71

0 Introduction

0.1 Basic background

1. PostgreSQL és una base de dades.
 - A. Cert
 - B. Fals
2. ¿Quin NO podem considerar un sinònim?
 - A. Taula
 - B. Registre
 - C. Dataset
 - D. Fitxer
 - E. Relació
 - F. Classe
3. ¿Quin NO podem considerar un sinònim?
 - A. Característica
 - B. Individu
 - C. Registre
 - D. Tupla
 - E. Fila
 - F. Instància
4. ¿Quin NO podem considerar un sinònim?
 - A. Característica
 - B. Camp
 - C. Atribut
 - D. Registre
 - E. Columna
5. ¿Quin NO és un dels subsistemes d'una organització?
 - A. Producció
 - B. Decisió
 - C. Informació
 - D. Econòmic
6. ¿A quin dels subsistemes d'una organització dona suport el subsistema d'informació?
 - A. Només al de producció
 - B. Només al de decisió
 - C. A tots dos
 - D. A cap dels dos

0.2 Kinds of systems

7. ¿Quin és un tipus sistema d'informació operational?
 - A. DW
 - B. OLTP
 - C. Machine Learning
 - D. OLAP
8. ¿Quin tipus de sistema d'informació és orientat a temes?
 - A. Operational
 - B. Decisional
9. ¿Quin tipus de sistema d'informació té més usuaris?
 - A. Decisional
 - B. Operational
10. ¿Les consultes de quin tipus de sistema d'informació accedeixen, en general, més tuples?
 - A. Operational
 - B. Decisional
11. ¿Quin tipus de sistema d'informació utilitza menys fonts de dades?
 - A. Decisional
 - B. Operational
12. ¿Quin tipus de sistema d'informació utilitza accedeix dades atòmiques?
 - A. Operational
 - B. Decisional
13. ¿Quin tipus de sistema d'informació utilitza principalment dades històriques?
 - A. Decisional
 - B. Operational
14. Un sistema d'informació decisional es considera ...
 - A. ... *Read-only*
 - B. ... *Write-only*
 - C. ... *Read-write*
15. ¿Quin tipus de sistema d'informació utilitza transaccions?
 - A. Operational
 - B. Decisional
16. ¿Per a quin tipus de sistemes d'informació es rellevant el seu rendiment (*performance*)?
 - A. Només operationals
 - B. Només decisionals

- C. Tots dos
 - D. Cap dels dos
17. ¿Quin tipus de sistema d'informació ha de gestionar un volum de dades més gran?
- A. Decisional
 - B. Operational
18. El disseny dels sistemes decisionals es fa segons ...
- A. Problemes
 - B. ... Funcionalitats
 - C. ... Estats
 - D. ... Temes

0.3 Design steps

19. ¿Quin NO és una de les tres fases principals del disseny de bases de dades?
- A. Lògic
 - B. Captura de requisits
 - C. Físic
 - D. Conceptual
20. El disseny de la base de dades relacional per a un sistema d'informació segueix els mateixos tres passos, independentment que aquest sigui operacional o decisional.
- A. Cert
 - B. Fals
21. La diferència en la fase de disseny conceptual d'un sistema operacional i un decisional és que l'operacional utilitza qualsevol estructura de classes i interrelacions, mentre que el decisional utilitza una molt concreta.
- A. Cert
 - B. Fals
22. La diferència en la fase de disseny lògic d'un sistema operacional i un decisional és que el decisional utilitza qualsevol estructura de taules i restriccions d'integritat, mentre que l'operacional utilitza una molt concreta.
- A. Cert
 - B. Fals
23. ¿Què és el més rellevant pel disseny de la base de dades?
- A. Diagrames de seqüència

- B. Diagrama de classes
 - C. Casos d'ús
 - D. Diagrama d'estats
24. ¿En que es basa el disseny de la base de dades?
- A. Només amb les dades
 - B. Només amb els processos
 - C. En les dues coses
 - D. En cap de les dues coses

0.4 Kinds of models

25. És fonamental que l'esquema lògic l'entenguin els usuaris del sistema.
- A. Cert
 - B. Fals
26. ¿Quin dels models ha de ser més expressiu?
- A. El conceptual
 - B. El lògic
 - C. Els dos igual
 - D. No importa
27. ¿En quina dècada va aparèixer la primera versió de SQL?
- A. 70s
 - B. 80s
 - C. 90s
 - D. 00s

1 Relational Translation - Difficulties, Criteria and Tools

1.1 Design Steps

28. El mètode de disseny que utilitzem NO depèn de la mida de la companyia.
- A. Cert
 - B. Fals
29. El mètode de disseny que utilitzem és diferent si ja existeix una base de dades prèviament o no.
- A. Cert
 - B. Fals
30. ¿Quina NO és una alternativa de disseny?
- A. Automàtica
 - B. Integració de vistes
 - C. Manual
 - D. Definició de procediments
31. ¿Quina és la fase de disseny més tardana?
- A. Quantificació de volums i freqüències
 - B. Consideracions respecte a temps de resposta, concurrència, recuperació y seguretat
 - C. Control de rendiment (monitorització i pla d'accés)
 - D. Afinació (*Tuning*)

1.2 Interpretation

32. Per ajudar-nos a entendre millor les multiplicitats de les associacions, podem dibuixar algunes instàncies i les seves interrelacions.
- A. Cert
 - B. Fals
33. L'esquema lògic de la base de dades, només hauria de permetre instanciacions correctes de l'esquema conceptual.
- A. Cert
 - B. Fals

1.3 Semantic Relativism

34. ¿Què NO és relativisme semàntic?
- A. Dos atributs que representen el mateix valor, però en unitats de mesura diferents
 - B. Dues classes del mateix concepte, però amb noms diferents
 - C. Dues taules amb dades equivalents, però amb noms diferents

- D. Dos procediments emmagatzemats que fan el mateix, però en llenguatges de programació diferents
35. Un atribut d'una classe, es pot representar també mitjançant una associació d'aquesta amb una segona classe.
- A. Cert
 - B. Fals
36. Una especialització mai pot ser equivalent a una agregació.
- A. Cert
 - B. Fals
37. Parlem de relativisme semàntic només quan els dos (o més) esquemes representen exactament la mateixa realitat.
- A. Cert
 - B. Fals
38. Parlem de relativisme semàntic només quan els dos (o més) esquemes contenen exactament el mateix nombre de taules.
- A. Cert
 - B. Fals

1.4 Null Values

39. El valor *null* és un dels del domini de l'atribut.
- A. Cert
 - B. Fals
40. Utilitzar un zero és equivalent a utilitzar un valor *null*, sempre que el primer no pertanyi al domini de l'atribut.
- A. Cert
 - B. Fals
41. ¿En quin cas NO s'haurien d'utilitzar valor nulls?
- A. Per a poder inserir una tupla amb un valor desconegut
 - B. Per no haver de tractar casos especials en les funcions d'agregació
 - C. Per a poder inserir una tupla amb un valor que mai pot tenir
 - D. Per a afegir una columna a una taula que no està buida
42. El problema d'utilitzar valors *null* és que malbaratem espai.
- A. Cert

- B. Fals
43. ¿Quina afirmació es certa?
- A. El valor null dividit per zero dona error
 - B. Un null igualat amb ell mateix (“NULL=NULL”) avalua cert
 - C. Un valor null ocupa el mateix espai que qualsevol altra valor
 - D. Cap de les anteriors afirmacions és certa
44. Les taules de veritat de lògica binària són un subconjunt de les de lògica ternària, on simplement eliminem les files i columnes que fan referència a “Desconegut”.
- A. Cert
 - B. Fals
45. ¿Quina afirmació és certa?
- A. Una tupla que té un valor *null* a la clau forana viola la restricció d'integritat
 - B. Una tupla que avalua el predicat d'un CHECK a “desconegut” viola la restricció d'integritat
 - C. Una tupla que avalua el predicat del WHERE a “desconegut” surt al resultat de la consulta
 - D. Cap de les anteriors afirmacions és certa
46. Si a l'entrada de l'operació algebraica d'unió hi ha dues files que només tenen valor *null* a tots els seus atributs, a la sortida tindrà només una fila.
- A. Cert
 - B. Fals
47. Si a l'entrada de l'operació algebraica d'intersecció hi ha dues files (una a cada taula) que només tenen valor *null* a tots els seus atributs, a la sortida no apareixerà aquesta fila.
- A. Cert
 - B. Fals
48. Si a l'entrada de l'operació algebraica de diferència hi ha dues files (una a cada taula) que només tenen valor *null* a tots els seus atributs, a la sortida no apareixerà aquesta fila.
- A. Cert
 - B. Fals
49. Si com a resultat d'una operació algebraica de projecció queda més d'una fila que només té valor *null* a tots els seus atributs, a la sortida apareixerà aquesta fila tantes vegades com la tingues a l'entrada.
- A. Cert
 - B. Fals
50. ¿Quina de les següents expressions dona error en SQL estàndard?
- A. 0/NULL
 - B. NULL/0
 - C. NULL/NULL

- D. Cap d'elles dona error
51. El GROUP BY genera un grup per a cada valor *null* que troba.
- A. Cert
 - B. Fals
52. El UNIQUE permet tenir més d'un valor nul.
- A. Cert
 - B. Fals
53. Sempre puc implementar una diferència indistintament amb un “NOT IN” o un “NOT EXISTS”.
- A. Cert
 - B. Fals

1.5 Generalization/Specialization

54. ¿Quin NO és un tipus de Generalització/Especialització?
- A. Disjoint
 - B. Complete
 - C. Overlapping
 - D. Imbalanced
55. En qualsevol dels tres tipus d'implementació de Generalització/Especialització posem un atribut discriminant.
- A. Cert
 - B. Fals
56. En cap dels tres tipus d'implementació de Generalització/Especialització posem claus foranes.
- A. Cert
 - B. Fals
57. ¿Quina de les tres implementacions de Generalització/Especialització NO genera mai nuls ni repeteix valors?
- A. Creació d'una única taula corresponent a la superclasse
 - B. Creació d'una taula per cada classe (tant per a la superclasse com per a les subclasses)
 - C. Creació d'una taula per cada subclasse, però no per a la superclasse
 - D. Cap de les tres, totes poden generar problemes depenent del tipus de Generalització/Especialització
58. Quan implementem una Generalització/Especialització en un SGBD relacional, sempre cal fer una *join* per consultar juntes totes les dades de la superclasse.
- A. Cert

- B. Fals
59. Quan implementem una Generalització/Especialització en un SGBD relacional, sempre cal utilitzar una *outer join* per consultar juntes totes les dades de qualsevol de les subclasses.
- A. Cert
B. Fals
60. ¿Quin dels tres tipus de *join* NO genera mai valors *null*?
- A. *Full outer join*
B. *Right outer join*
C. *Left outer join*
D. Tots tres en poden generar
61. Si una consulta té només dues taules al *FROM*, podem intercanviar *LEFT* per *RIGHT* i obtenir el mateix resultat, simplement canviant a la vegada l'ordre de les taules.
- A. Cert
B. Fals
62. La *FULL OUTER JOIN* de dues taules permet obtenir el mateix resultat que una *LEFT OUTER JOIN* o una *RIGHT OUTER JOIN*, simplement afegint un cert predicat al *WHERE*.
- A. Cert
B. Fals
63. La *FULL OUTER JOIN* de dues taules és equivalent a fer la unió d'una *LEFT OUTER JOIN* i una *RIGHT OUTER JOIN*.
- A. Cert
B. Fals
64. Quan utilitzem la implementació de l'herència en PostgreSQL, ¿Quina de les següents afirmacions és certa?
- A. Només podem inserir dades a la taula de la subclasse
B. Només podem inserir dades a la taula de la superclasse
C. Podem inserir dades indistintament a qualsevol de les dues taules i tindrem de mateix resultat
D. Podem inserir dades a qualsevol de les dues taules, però no tindrem del mateix resultat

1.6 Surrogates

65. ¿Quina NO és una raó per a crear un *surrogate*?
- A. No existeix una clau externa
B. El atributs de la clau externa canvien molt sovint
C. La clau externa requereix massa espai

- D. Totes ho son
66. Un atribut tipus *SERIAL* sempre tindrà valors consecutius tret que haguem esborrat alguna fila de la taula.
- A. Cert
 - B. Fals
67. Una *SEQUENCE* de PostgreSQL es pot utilitzar a taules diferents.
- A. Cert
 - B. Fals

2 Relational Translation - Relationships

2.1 Constraints

68. ¿Com podem implementar les restriccions d'integritat?
- A. Dins del CREATE TABLE
 - B. Amb *Persistent Stored Modules*
 - C. Amb JDBC
 - D. Amb qualsevol de les anteriors
69. ¿La clau primària d'una taula sempre genera automàticament un índex B+ associat?
- A. Cert
 - B. Fals
70. ¿Les claus alternatives no tenen una clàusula pròpia a l'estàndard per a crear-les?
- A. Cert
 - B. Fals
71. La millor manera per a resoldre un *deadlock* provocat per claus foranes entre dues taules, és sempre crear una de les taules sense clau forana i afegir-li a posteriori.
- A. Cert
 - B. Fals
72. Un *deadlock* provocat per claus foranes entre dues taules, es pot resoldre creant una tercera taula.
- A. Cert
 - B. Fals
73. ¿Quina és la millor opció per inserir tuples quan tenim un *deadlock* provocat per claus foranes entre dues taules?
- A. Diferir la comprovació de les claus foranes
 - B. Desactivar una de les claus foranes i tornar a activar-la després
 - C. Esborrar una de les claus foranes i tornar a crear-la després
 - D. Cap de les anteriors
74. ¿Què perdem si esborrem una taula que només té l'atribut identificador i una clau forana que l'apunta?
- A. Les seves dades
 - B. Una part del disseny
 - C. La restricció d'integritat
 - D. La seva informació

2.2 Binary Associations

75. ¿Què és menys important en una associació binària quan fem el disseny conceptual de la base de dades?
- A. Si la multiplicitat màxima de cada costat
 - B. Si la multiplicitat mínima de cada costat
 - C. El nom de l'associació
 - D. La direcció de navegació
76. ¿En quin tipus d'associació binària és totalment irrellevant si un dels costats admet zeros o no?
- A. *-*
 - B. 1-*
 - C. 1-1
 - D. Sempre és rellevant
77. ¿Quin tipus d'associació binària s'ha d'implementar sempre amb una taula pròpia (a més de les dues corresponents a les classes)?
- A. *-*
 - B. 1-*
 - C. 1-1
 - D. Totes les anteriors
78. ¿En quin tipus d'associació binària podem triar a quina taula posem la clau forana?
- A. *-*
 - B. 1-*
 - C. 1-1
 - D. Totes les anteriors
79. Independentment de les multiplicitats d'una associació binària, sempre hi ha una opció per evitar que es puguin generar valors *null*.
- A. Cert
 - B. Fals
80. El millor és sempre triar la implementació d'una associació binària que no genera mai valors *null*.
- A. Cert
 - B. Fals
81. En el model relacional clàssic, ¿en presència de quin tipus d'associació binària podem fusionar les dues taules en una?
- A. *-*
 - B. 1-*

- C. 1-1
- D. Totes les anteriors

82. ¿Quin NO és un criteri per a triar la clau primària entre les alternatives que hi hagin?
- A. Freqüència d'ús a les consultes
 - B. Freqüència de canvi
 - C. Espai requerit
 - D. Nombre de dependències funcionals

2.3 Reflexive Associations

83. ¿Quina de les següents afirmacions respecte a les associacions reflexives és falsa?
- A. Poden ser simètriques o no
 - B. Poden generar cadenes amb infinites instàncies
 - C. En podem trobar amb qualsevol multiplicitat
 - D. Gairebé sempre tenen zeros
84. Una associació reflexiva simètrica només pot tenir multiplicitat 1-1.
- A. Cert
 - B. Fals
85. En una associació reflexiva simètrica, cal guardar sempre a la base de dades les dues versions de la parella.
- A. Cert
 - B. Fals
86. Per tal de garantir la simetria d'una associació binària, sempre haurem d'utilitzar *triggers*.
- A. Cert
 - B. Fals

2.4 N-ary Associations

87. La implementació d'una associació ternària genera ...
- A. ... una clau candidata
 - B. ... dues claus candidates
 - C. ... una o dues claus candidates
 - D. ... una, dues o tres claus candidates
88. Hi ha casos en que podem implementar una associació ternària amb simplement una clau forana (sense cap taula intermitja).
- A. Cert
 - B. Fals
89. La implementació d'una associació ternària en un SGBD relacional mai genera valors *null*.
- A. Cert
 - B. Fals

2.5 Compound Aggregations

90. Una classe pot se part de dues amb una agregació composta.
- A. Cert
 - B. Fals
91. En una agregació composta, tots dos costats poden tenir un zero.
- A. Cert
 - B. Fals
92. Podem encadenar tantes agregacions compostes com vulguem.
- A. Cert
 - B. Fals
93. Una clau forana de dos atributs és equivalent a dues claus foranes d'un atribut cadascuna.
- A. Cert
 - B. Fals

2.6 Association Classes

94. Quan en l'esquema conceptual trobem una associació binària, independentment de la multiplicitat, sempre podem transformar-la en una classe que tingui el mateix nom que l'associació i dues associacions amb les classes originals.
- A. Cert
 - B. Fals
95. A les associacions binàries, només les *-* poden tenir atributs propis.
- A. Cert
 - B. Fals
96. Si una associació binària té un atribut, el posem ...
- A. ... a la taula del costat que té zero
 - B. ... a la taula del costat que no té zero
 - C. ... a la taula on posem la clau forana
 - D. ... a qualsevol de les taules
97. Una associació dona lloc realment a una classe associativa quan ...
- A. ... té molts atributs
 - B. ... correspon a una associació *-*
 - C. ... és part d'una agregació composta
 - D. ... té identificador

2.7 Multivalued Attributes

98. Quan un atribut multivaluat pot tenir molts valors, es millor emmagatzemar-ho per columnes.
- A. Cert
 - B. Fals
99. Un atribut multivaluat emmagatzemat per files mai genera valors *null*.
- A. Cert
 - B. Fals
100. Un atribut multivaluat emmagatzemat per columnes genera un únic accés a disc per recuperar tots els valors.
- A. Cert
 - B. Fals
101. Un atribut multivaluat emmagatzemat per columnes ocupa menys espai.
- A. Cert
 - B. Fals
102. Un atribut multivaluat emmagatzemat per files és més difícil d'agregar.
- A. Cert
 - B. Fals
103. Un atribut multivaluat emmagatzemat per files permet un grau més alt de concurrència.
- A. Cert
 - B. Fals

2.8 Example of Translation

104. Sempre podem implementar totes les restriccions d'integritat corresponents a les multiplicitats "1" i "*" de les associacions binàries amb només claus primàries, foranes i "*NOT NULL*".
- A. Cert
 - B. Fals

3 Normalization

3.1 Anomalies

105. Les anomalies a les que fa referència la normalització no afecten a les consultes.
- A. Cert
 - B. Fals
106. ¿Quina NO és una conseqüència de les anomalies que motiven la teoria de la normalització?
- A. Algunes dades es poden perdre sense voler
 - B. Un únic canvi pot provocar moltes modificacions a la base de dades
 - C. Es generen joins innecessàries
 - D. No podem inserir una certa dada de forma independent
107. La normalització formalitza les propietats que ha de complir un disseny de bases de dades relacionals per a ser considerat de qualitat.
- A. Cert
 - B. Fals
108. ¿Quin és el propòsit fonamental de la normalització?
- A. Cada relació correspon a una classe
 - B. Cada relació correspon a una entitat
 - C. Cada relació correspon a una funcionalitat
 - D. Cada relació correspon a un concepte semàntic

3.2 Functional Dependencies

109. El producte cartesià de dos conjunts és també un tipus de relació, però no funcional.
- A. Cert
 - B. Fals
110. Una clau primària correspon a ... entre la instància i el valor
- A. ... una funció exhaustiva
 - B. ... una funció bijectiva
 - C. ... una funció injectiva
 - D. ... cap de les anteriors
111. Una clau primària i un altre atribut de la mateixa relació podem estar relacionats amb ...
- A. ... una funció injectiva
 - B. ... una funció bijectiva
 - C. ... una funció exhaustiva

D. ... qualsevol de les anteriors

112. Una dependència funcional de X a Y ($\{X\} \rightarrow \{Y\}$) vol dir que si sabem el valor de X, podem saber el de Y.

A. Cert

B. Fals

113. Una dependència funcional plena és una dependència funcional amb un únic atribut a la part dreta.

A. Cert

B. Fals

3.3 Normal Forms

114. El principal propòsit de les formes normals del model relacional és millorar el rendiment (és a dir, temps de resposta) del sistema.

A. Cert

B. Fals

115. Una relació està en 1NF si tots els seus atributs són atòmics (és a dir, indivisibles).

A. Cert

B. Fals

116. ¿Quin dels següents tipus de dades és atòmic?

A. JSON

B. Matriu

C. Tots dos ho són

D. Cap dels dos ho és

117. Una relació està en 2NF si totes les dependències funcionals cap als atributs que no formen part d'una clau són plenes.

A. Cert

B. Fals

118. Si una relació que no està en 2FN i volem que ho estigui, el que hem de fer és dividir els seus atributs en subconjunts disjunts tant com calgui fins que ho estigui.

A. Cert

B. Fals

119. Una relació està en 3NF si cap dels seus atributs està a la part esquerra d'una dependència funcional.

A. Cert

B. Fals

120. Si una relació que no està en 3FN i volem que ho estigui, el que hem de fer és dividir els seus atributs en subconjunts disjunts tant com calgui fins que ho estigui, però deixant en cada divisió com clau forana entre les parts el determinant que provoca la violació de la forma normal.

A. Cert

B. Fals

121. ¿Quines formes normals tenen excepció?

A. 1NF i 2NF

B. 2NF i 3NF

C. 1NF i 3NF

D. 1NF, 2NF, 3NF

122. BCNF no permet tenir claus alternatives.

A. Cert

B. Fals

123. La BCNF es va definir perquè 2NF i 3NF no garanteixen l'absència d'anomalies.

A. Cert

B. Fals

124. La normalització fins a BCNF és única.

A. Cert

B. Fals

3.4 Automatic Normalization

125. ¿Quina NO és una de les regles d'Armstrong?

A. Distributivitat

B. Pseudo-transitivitat

C. Adició

D. Projectabilitat/Descomposició

E. Augmentativitat

F. Transitivitat

G. Reflexivitat

126. La clausura de dependències funcionals conté totes les dependències implícites al disseny.

A. Cert

B. Fals

127. ¿Què NO puc deduir a partir de la clausura de dependències funcionals?

A. Tot el conjunt de claus candidates

- B. Si un esquema es correcte
 - C. Si dos esquemes relacionals són equivalents
 - D. Si una dependència funcional és certa o no
128. Amb l'algorisme de normalització d'anàlisi es poden perdre dependències funcionals.
- A. Cert
 - B. Fals
129. Si executem l'algorisme de normalització d'anàlisi diverses vegades, podem obtenir resultats diferents.
- A. Cert
 - B. Fals
130. L'algorisme de normalització d'anàlisi NO garanteix que al final complim BCNF.
- A. Cert
 - B. Fals
131. L'algorisme de normalització d'anàlisi tracta de trobar la "relació universal".
- A. Cert
 - B. Fals

3.5 Examining Normalization

132. No tots els casos es poden normalitzar fins a BCNF.
- A. Cert
 - B. Fals
133. Donades les dependències funcionals, la normalització es pot automatitzar totalment.
- A. Cert
 - B. Fals
134. Un esquema normalitzat és millor que un que no ho està, perquè el primer conté redundàncies que el segon no té.
- A. Cert
 - B. Fals
135. ¿Quina NO és una raó per a desnormalitzar?
- A. Quan no s'esperen canvis a les dades
 - B. Quan les taules s'haurien d'ajuntar (*join*) molt sovint
 - C. Quan la consistència de les dades es pot garantir d'alguna altra manera

D. Quan la base de dades és molt petita

136. Donada la seqüència de conjunts d'esquemes tals que compleixen una certa dependència funcional: E_{1NF} , E_{2NF} , E_{3NF} , E_{BCNF} i E_{4NF} (per exemple, E_{3NF} representa tots el esquemes que compleixen 3NF). Si considerem qualsevol parella E_X - E_Y tal que E_X apareix abans que E_Y en aquesta seqüència, podem afirmar que:

- A. $E_X \cap E_Y = \emptyset$ (no hi ha cap esquema que compleixi X i Y alhora)
- B. $E_X \supseteq E_Y$ (tots els que compleixen Y , també compleixen X)
- C. $E_X \subseteq E_Y$ (tots els que compleixen X , també compleixen Y)
- D. Cap de les anteriors

137. Qualsevol relació està en NF^2 .

- A. Cert
- B. Fals

3.6 4NF

138. Una dependència multivaluada és un tipus de dependència funcional.

- A. Cert
- B. Fals

139. Si una relació està en 4NF ho està també en BCNF.

- A. Cert
- B. Fals

140. La 4NF no permet tenir dependències multivaluades.

- A. Cert
- B. Fals

141. Per normalitzar una relació que no estigui en 4FN, hem de fer el producte cartesià del atributs que l'estiguin violant.

- A. Cert
- B. Fals

4 Data Warehousing and OLAP

4.1 Operational and Decisional Comparison

142. ¿Quin tipus de sistema NO forma part del suport a la presa de decisions?
- A. Machine Learning
 - B. ETL
 - C. Reporting
 - D. OLAP
 - E. ERP
 - F. Data Warehouse
143. ¿Quin tipus de sistema té un us més repetitiu?
- A. Operacional
 - B. Decisional
 - C. Els dos similar
144. ¿Quin tipus de sistema té més usuaris?
- A. Operacional
 - B. Decisional
 - C. Els dos similar
145. ¿En quin tipus de sistema una consulta acostuma a accedir més tuples?
- A. Operacional
 - B. Decisional
 - C. Els dos similar
146. ¿Quin tipus de sistema té més dades històriques?
- A. Operacional
 - B. Decisional
 - C. Els dos similar
147. ¿Quin tipus de sistema té més operacions de modificació?
- A. Operacional
 - B. Decisional
 - C. Els dos similar
148. ¿Quin tipus de sistema utilitza transaccions?
- A. Operacional
 - B. Decisional
 - C. Els dos
 - D. Cap dels dos
149. ¿Quin tipus de sistema té consultes més complexes?
- A. Operacional

- B. Decisional
 - C. Els dos similar
150. ¿En quin tipus de sistema és més important el seu rendiment?
- A. Operacional
 - B. Decisional
 - C. En els dos
151. ¿Quin tipus de sistema emmagatzema més dades?
- A. Operacional
 - B. Decisional
 - C. Els dos similar

4.2 Data Warehouse Definition

152. ¿Quina NO és una de les quatre característiques d'un magatzem de dades?
- A. No volàtil
 - B. Transaccional
 - C. Històric
 - D. Integrat
 - E. Orientat a temes
153. ¿Quin tipus de temps ha de gestionar un magatzem de dades?
- A. Temps de Transacció
 - B. Temps de Validesa
 - C. Tots dos
 - D. Cap dels dos
154. ¿Quina característica té un magatzem de dades?
- A. Orientat a temes
 - B. Orientat a la funcionalitat
 - C. Orientat a decisions
 - D. Orientat a transaccions
155. ¿Que un magatzem de dades sigui “*time variant*” vol dir que el software associat està en permanent evolució?
- A. Cert
 - B. Fals
156. ¿Que un magatzem de dades sigui “*non-volatile*” vol dir que mai s'esborra cap dada?

- A. Cert
- B. Fals

157. ¿Que un magatzem de dades sigui “*integrated*” vol dir que els seus components de programari interactuen com si fos un sistema monolític?

- A. Cert
- B. Fals

4.3 Data Warehousing Architectures

158. ¿Quina diferència hi ha entre una arquitectura d’un sol nivell i una de dos d’emmagatzemament de dades (*data warehousing*)?

- A. La d’un nivell no té cap base de dades
- B. La d’un nivell no té cap base de dades operacional
- C. La d’un nivell no té cap base de dades decisional
- D. Cap de les anteriors

159. Una arquitectura d’un sol nivell d’emmagatzemament de dades (*data warehousing*), genera un magatzem de dades virtual.

- A. Cert
- B. Fals

160. En una arquitectura de dos nivells d’emmagatzemament de dades (*data warehousing*), les dades es poden replicar com a màxim dues vegades.

- A. Cert
- B. Fals

161. ¿Quin tipus de tasca NO fa un procés ETL?

- A. Carregar les dades a una base de dades
- B. Treure dades de les fonts
- C. Encapsular l’accés a les dades
- D. Netejar les dades

162. Les bases de dades decisionals admeten redundàncies.

- A. Cert
- B. Fals

163. Les bases de dades decisionals sempre actualitzen les dades en temps real.

- A. Cert
- B. Fals

164. Les bases de dades decisionals NO necessiten control de concurrència.

- A. Cert
- B. Fals

4.4 OLAP Definition

165. ¿Quina NO es considera una limitació dels fulls de càlcul per analitzar dades multidimensionals?
- A. No gestiona metadades
 - B. No permeten operacions d'àlgebra linial
 - C. Permet un número limitat de cel·les
 - D. La posició de les dades limita certes operacions
 - E. No gestiona jerarquies d'agregació
166. ¿Quina NO és una característica de les eines OLAP?
- A. Integra dades
 - B. Respon consultes ràpidament
 - C. Gestiona dades multidimensionals
 - D. Serveix per a fer anàlisi de dades
 - E. Permet compartir dades
167. Les eines OLAP guarden les dades en una estructura de dades física en forma de cub n-dimensional.
- A. Cert
 - B. Fals

4.5 OLAP Operations

168. Un cub de dades es una metàfora per a una taula estadística.
- A. Cert
 - B. Fals
169. Una taula relacional és equivalent a un cub de dades.
- A. Cert
 - B. Fals
170. ¿Quina NO és una operació de les eines OLAP?
- A. *Slice*
 - B. *Dice*
 - C. *Drill-down*
 - D. *Roll-up*
 - E. *Cube*
171. Les operacions de “*Slice&Dice*” són seleccions sobre les dades.
- A. Cert
 - B. Fals
172. Les operacions de “*Roll-up*” i “*Drill-Down*” indiquen agrupacions de dades.
- A. Cert
 - B. Fals

4.6 Conceptual Multidimensional Schema

173. ¿Quin avantatge de la modelització de dades operacionals ens és útil també per modelitzar dades decisionals?

- A. Facilita la comprensió del domini per part dels usuaris finals
- B. Resulta en sistemes molt eficients quan tenen canvis freqüents
- C. Redueix la quantitat de dades redundants
- D. Elimina la necessitat de modificar molts registres per un sol canvi

174. ¿Què busquem quan fem un esquema en estrella?

- A. Posar el focus en un tema concret
- B. Simplificar l'esquema conceptual
- C. Tots dos
- D. Cap dels dos

175. ¿Quina multiplicitat permet un esquema en estrella entre el fet i les dimensions?

- A. 1-1
- B. 1-*
- C. *-*
- D. Totes les anteriors

176. Un esquema en estrella té un únic fet.

- A. Cert
- B. Fals

177. Un esquema en estrella no pot tenir més de quatre dimensions.

- A. Cert
- B. Fals

4.7 Logical Multidimensional Schema

178. ¿Quins nivells té una eina ROLAP?

- A. Emmagatzemament i consulta
- B. Consulta i traducció
- C. Emmagatzemament i traducció
- D. Tots els anteriors

179. ¿Quin tipus de SGBD utilitzen les eines ROLAP?

- A. NOSQL
- B. Relacional
- C. Multidimensional
- D. Cap

180. Cada SGBD té la seva pròpia llibreria ROLAP.

- A. Cert

- B. Fals
181. El principal problema de les eines ROLAP és que generen massa *joins*.
- A. Cert
- B. Fals
182. Els sistemes OLAP i OLTP implementats sobre SGBDs relacionals segueixen les mateixes tres fases de disseny.
- A. Cert
- B. Fals
183. ¿Quantes claus primàries té la taula de fets d'un esquema en estrella?
- A. Tantes com dimensions menys una
- B. Tantes com dimensions
- C. Una
- D. Un número indeterminat
184. ¿Quants atributs té la clau primària de la taula de fets d'un esquema en estrella?
- A. Un
- B. Tants com dimensions menys una
- C. Tants com dimensions
- D. Un número indeterminat
185. Una *Cube-Query* fa tantes *joins* com dimensions tingui el cub més una.
- A. Cert
- B. Fals
186. Una *Cube-Query* sobre una eina ROLAP retorna una taula relacional.
- A. Cert
- B. Fals

4.8 SQL Extension

187. ¿De quin tipus de valors faciliten el càlcul les clàusules "GROUPING SETS"?
- A. Agregats
- B. Atòmics
- C. Marginals
- D. Derivats
188. ¿Quin NO és un possible significat del valor *null*?
- A. Desconegut
- B. Agregat

- C. Integrat
- D. Inexistent

189. Qualsevol consulta que utilitzi *ROLLUP* i/o *CUBE*, sempre es pot rescriure utilitzant només *GROUPING SETS*.

- A. Cert
- B. Fals

190. Sempre que es pugui, es millor utilitzar *ROLLUP* i/o *CUBE*, en lloc de *GROUPING SETS*, perquè facilita l'optimització de la consulta.

- A. Cert
- B. Fals

5 NOSQL

5.1 Motivation

191. Els SGBDs relacionals permeten implementar qualsevol tipus de sistema d'informació de forma eficient.
- A. Cert
 - B. Fals
192. Els sistemes NOSQL són *schemaless*.
- A. Cert
 - B. Fals
193. Els sistemes NOSQL garanteixen les restriccions d'integritat, tal com fan els sistemes relacionals.
- A. Cert
 - B. Fals
194. *Impedance mismatch* vol dir que el format de les dades al disc i a la memòria és diferent.
- A. Cert
 - B. Fals

5.2 Schema Definition

195. El fet de tenir unes instàncies dins d'unes altres trenca la 1NF.
- A. Cert
 - B. Fals
196. ¿Quina NO és una conseqüència d'acceptar la variabilitat de l'esquema?
- A. Es guanya en flexibilitat
 - B. Es perd semàntica
 - C. Es perd el principi d'independència de dades
 - D. Es guanya eficiència en les consultes
197. La independència física, d'acord amb l'arquitectura ANSI/SPARC, garanteix que un canvi en l'esquema intern no afectarà a les taules de la base de dades.
- A. Cert
 - B. Fals
198. La independència lògica, d'acord amb l'arquitectura ANSI/SPARC, garanteix que un canvi en les vistes no afectarà a les taules de la base de dades.
- A. Cert
 - B. Fals
199. La independència lògica, d'acord amb l'arquitectura ANSI/SPARC, garanteix que un canvi en una taula no afectarà a cap vista de la base de dades, tret que el canvi afecti els atributs o taules utilitzats en la vista.

- A. Cert
- B. Fals

5.3 Storing Arrays

200. Una SGBD relacional pot contenir una base de dades que segueixi a la vegada el model relacional i corelacional.

- A. Cert
- B. Fals

201. En PostgreSQL, per canviar el valor d'una posició concreta d'un *array*, hem de reassignar tot l'*array* complet a la fila corresponent.

- A. Cert
- B. Fals

202. Un *array* en PostgreSQL té un nombre fix de valors, indicat en la seva declaració.

- A. Cert
- B. Fals

203. Un *array* en PostgreSQL pot contenir valors *nulls*.

- A. Cert
- B. Fals

204. Un atribut multivaluat guardat en un *array* generarà tans accessos a disc com elements tingui l'*array*.

- A. Cert
- B. Fals

205. Un atribut multivaluat guardat en un *array* ocuparà més espai de disc que guardant-ho per files, però menys que fent-ho per columnes.

- A. Cert
- B. Fals

206. En general, la funció per calcular agregats a partir d'un *array*, l'ha de definir l'usuari.

- A. Cert
- B. Fals

207. PostgreSQL disposa de funcions booleanes específiques que permeten definir fàcilment restriccions d'integritat sobre cadascun dels elements d'un *array*.

- A. Cert
- B. Fals

208. El nivell de concurrència que permet guardar un atribut multivaluat en un *array* és el mateix que si el guardem per columnes.

- A. Cert
- B. Fals

5.4 Semi-structured Database Model

209. La principal diferència entre un gestor de documents i un de parelles clau-valor és que el primer aprofita l'estructura dels documents per a permetre la definició d'índexs secundaris.

- A. Cert
- B. Fals

210. XML és un model de base de dades semi-estructurat.

- A. Cert
- B. Fals

211. JSON és un model de base de dades semi-estructurat.

- A. Cert
- B. Fals

212. Quan tenim documents JSON, els hem de guardar en un SGBD relacional utilitzant el tipus de dades corresponent.

- A. Cert
- B. Fals

213. Podem tenir un document JSON que compleixi la mateixa propietat de la BCNF (és a dir, cada determinant determina per ell mateix tots els atributs del document, ja sigui directa o indirectament).

- A. Cert
- B. Fals

214. Podem tenir claus primàries en documents JSON.

- A. Cert
- B. Fals

215. Podem tenir claus foranes en documents JSON.

- A. Cert
- B. Fals

216. El principal propòsit de niuar diferents instàncies en un mateix document JSON és evitar *joins*.

- A. Cert
- B. Fals

217. En un gestor de documents, podem definir l'esquema que aquests han de seguir.

- A. Cert
- B. Fals

5.5 JSON Data Type

218. Un gestor de documents guarda sempre al disc exactament el mateix que posa a la memòria.
- A. Cert
 - B. Fals
219. El tipus JSONB de PostgreSQL preserva l'ordre de les claus al document.
- A. Cert
 - B. Fals
220. El tipus JSONB de PostgreSQL elimina claus duplicades al document.
- A. Cert
 - B. Fals

6 Views

6.1 View Definition and Problems

221. ¿Quin NO és un dels quatre nivells de l'arquitectura ANSI/SPARC?
- A. Lògic
 - B. External
 - C. Conceptual
 - D. Internal
222. ¿En quin nivell de l'arquitectura ANSI/SPARC trobem més esquemes?
- A. Internal
 - B. Conceptual
 - C. External
 - D. Trobem a tots el mateix nombre
223. ¿Quina de les següents possibilitats no existeix en els SGBDs relacionals?
- A. Taules materialitzades
 - B. Vistes materialitzades
 - C. Taules no materialitzades
 - D. Vistes no materialitzades
224. La consulta que definiex una vista sempre està guardada al catàleg de la base de dades.
- A. Cert
 - B. Fals
225. ¿Quin és un exemple paradigmàtic de vistes materialitzades?
- A. DW
 - B. NOSQL
 - C. Tots dos
 - D. Cap dels dos
226. ¿Quina NO és una utilitat de les vistes?
- A. Comprovar restriccions d'integritat
 - B. Simplificar consultes
 - C. Simplificar el manteniment
 - D. Amagar dades
 - E. Simplificar esquemes complexos
227. ¿Quin NO és un dels quatre problemes associats a les vistes?
- A. *Query rewriting*
 - B. *Update through views*
 - C. *View expansion*
 - D. *View updating*

E. *View materialization*

228. ¿Quin problema aplica només a vistes no-materialitzades?

A. *Update through views*

B. *Query rewriting*

C. *View expansion*

D. *View updating*

E. Tots quatre problemes apliquen a qualsevol tipus de vista

229. ¿Quan es dona el problema de *View expansion*?

A. Quan modifiquem una taula

B. Quan consultem una vista

C. Quan modifiquem una vista

D. Quan consultem una taula

230. ¿Quan es dona el problema de *Query rewriting*?

A. Quan consultem una taula

B. Quan modifiquem una vista

C. Quan modifiquem una taula

D. Quan consultem una vista

231. ¿Quan es dona el problema de *View updating*?

A. Quan modifiquem una vista

B. Quan modifiquem una taula

C. Quan consultem una vista

D. Quan consultem una taula

232. ¿Quan es dona el problema de *Update through views*?

A. Quan consultem una taula

B. Quan consultem una vista

C. Quan modifiquem una vista

D. Quan modifiquem una taula

233. ¿Quan es varen introduir les vistes materialitzades a l'estàndard SQL?

A. Al seu inici

B. Al 1992

C. Al 2023

D. Encara no ho estan

234. ¿Quants dels tres problemes teòrics que tenen associats a les vistes materialitzades podem veure explícitament reflectits a la seva sentència de creació d'Oracle?

A. Cap

B. Un

C. Dos

D. Tres

6.2 View Expansion

235. Una vista sempre es pot expandir.
- A. Cert
 - B. Fals
236. Mai expandim vistes materialitzades.
- A. Cert
 - B. Fals
237. *View Expansion* consisteix simplement en substituir el nom de la vista que apareix a la consulta per la corresponent definició que podem trobar al catàleg de la base de dades.
- A. Cert
 - B. Fals
238. No podem expandir vistes dins d'una altra vista.
- A. Cert
 - B. Fals

6.3 Update Through Views

239. *Update Through Views* consisteix a utilitzar les vistes en comptes dels índexs per accedir més ràpid a les dades quan les hem de modificar.
- A. Cert
 - B. Fals
240. Els SGBDs permeten modificar una taula a través d'una vista independentment del predicat lògic que es posi al *WHERE* de la definició de la vista, sempre que aquest no contingui cap subconsulta.
- A. Cert
 - B. Fals
241. Els SGBDs permeten modificar una taula a través d'una vista definida sobre una altra vista si les dues definicions de les vistes compleixen, per separat, les condicions necessàries.
- A. Cert
 - B. Fals
242. Els SGBDs sempre permeten modificar una taula a través d'una vista que no conté cap altra taula.
- A. Cert
 - B. Fals
243. Els SGBDs sempre permeten modificar una taula a través d'una vista que tingui un agregat.
- A. Cert

B. Fals

244. Els SGBDs sempre permeten modificar una taula a través d'una vista que faci una *join*.

A. Cert

B. Fals

6.4 View Updating

245. *View Updating* consisteix a propagar els canvis que es produeixen a una taula a totes les vistes materialitzades definides sobre ella.

A. Cert

B. Fals

246. ¿Quan NO podem propagar els canvis d'una taula a les seves vistes materialitzades?

A. *Next <date>*

B. *On statement*

C. *On commit*

D. *On demand*

E. Podem propagar-los de totes les maneres anteriors

247. Només cal crear un *log* per fer manteniment de vistes incremental.

A. Cert

B. Fals

248. Hem de crear un únic *log* a cada taula que tingui vistes materialitzar a mantenir de forma incremental, independentment del seu nombre.

A. Cert

B. Fals

249. Fer el manteniment d'una vista materialitzada de forma incremental sempre és el més eficient.

A. Cert

B. Fals

250. Sempre es pot fer el manteniment d'una vista materialitzada de forma incremental si el *log* conté les dades adequades.

A. Cert

B. Fals

251. Sempre podem utilitzar una vista materialitzada per implementar una asserció.

A. Cert

B. Fals

252. Per implementar una asserció amb una vista materialitzada, cal que la vista sempre estigui buida.

A. Cert

B. Fals

6.5 Query Rewriting

253. *Query Rewriting* consisteix en que l'usuari rescrigui la seva consulta aprofitant les vistes materialitzades existents, en comptes de les taules.

- A. Cert
- B. Fals

254. En general, els SGBDs fan una cerca exhaustiva de totes les possibilitats de rescriptura d'una consulta emprant les vistes materialitzades que s'hagin definit amb anterioritat.

- A. Cert
- B. Fals

255. ¿Quin NO és un dels requisits per a poder rescriure una consulta en termes d'una vista materialitzada?

- A. Les taules utilitzades a la consulta han de ser un subconjunt de les que hi ha a la vista
- B. El predicat de la consulta ha d'estar subsumit pel de la vista
- C. El nivell d'agregació de la consulta ha de ser tan alt o més que el de la vista
- D. Els agregats de la consulta han de coincidir o ser calculables a partir dels de la vista

6.6 Materialized View Selection

256. Si, en general, utilitzar vistes materialitzades millora el rendiment de les consultes, NO és perquè ...

- A. ... generen menys accessos a disc
- B. ... tenen menys files
- C. ... tenen menys atributs
- D. ... ocupen menys espai
- E. ... generen menys contenció

257. Per decidir si val la pena materialitzar o no una certa consulta, cal tenir en compte ...

- A. ... la freqüència amb que s'executa
- B. ... l'espai que ocuparà el seu resultat
- C. ... la freqüència amb que es modifiquen les taules que accedeix
- D. ... totes les anteriors

258. Si el nostre SGBD no proporciona vistes materialitzades, podríem implementar-les nosaltres mateixos amb disparadors.

- A. Cert
- B. Fals

259. Les vistes materialitzades que podem materialitzar ve limitat per ...

- A. ... l'espai disponible en el disc
- B. ... el temps disponible per fer el seu manteniment

- C. ... totes les anteriors
 - D. ... cap de les anteriors
260. Només tenint en compte les diferents possibilitats d'agregació, el nombre de potencials vistes materialitzades que podem crear ja és exponencial.
- A. Cert
 - B. Fals
261. ¿Quina NO és una bona heurística per a triar quines vistes materialitzar en un esquema multidimensional?
- A. Materialitzar una vista si coincideix amb una consulta crítica
 - B. Materialitzar els nivells d'agregació més baixos
 - C. No materialitzar una vista si ja s'ha materialitzat una altra que és un ancestre proper en la jerarquia d'agregació
 - D. Materialitzar els nivells d'agregació més alts
 - E. Totes les heurístiques anteriors són bones
262. Hem de considerar com a vista candidata a materialitzar la que conté un “GROUP BY” que és la intersecció dels “GROUP BY” de dues consultes crítiques.
- A. Cert
 - B. Fals
263. Utilitzant un algorisme *greedy*, sempre tindrem el conjunt òptim de vistes materialitzades.
- A. Cert
 - B. Fals
264. Utilitzant un algorisme *greedy*, mai tindrem el conjunt òptim de vistes materialitzades.
- A. Cert
 - B. Fals
265. Si hem utilitzat un algorisme *greedy* per seleccionar-les, ja no hem de canviar mai el conjunt de vistes que hem decidit materialitzar.
- A. Cert
 - B. Fals

6.7 Example of Materialize View Selection

266. Podem estimar la cardinalitat del resultat d'una consulta amb agregats sobre la taula $T(a_1, \dots, a_n)$ com ...
- A. $|T|$
 - B. $dist(a_1) \cdot \dots \cdot dist(a_n)$
 - C. $\min(|T|, dist(a_1) \cdot \dots \cdot dist(a_n))$
 - D. $\max(|T|, dist(a_1) \cdot \dots \cdot dist(a_n))$

7 Physical Design

7.1 Tasks, Criteria and Difficulties

267. El disseny físic tracta d'adaptar l'esquema lògic a les particularitats d'un SGBD i una càrrega de treball (*workload*) concretes.

- A. Cert
- B. False

268. ¿Quina NO és una de les tasques bàsiques del disseny físic?

- A. Reconsiderar els requisits
- B. Escollir les estructures de dades
- C. Adaptar l'esquema lògic al SGBD
- D. Reconsiderar l'esquema relacional
- E. Testejar el rendiment

269. ¿Quin NO és un criteri per prendre decisions sobre el disseny físic?

- A. Benefici econòmic
- B. Disponibilitat
- C. Escalabilitat
- D. Simplicitat de l'administració
- E. Millora de rendiment
- F. Integritat

270. ¿Quina NO és una dificultat en el disseny físic?

- A. La xarxa
- B. Els usuaris
- C. La normalització
- D. Les imperfeccions del SGBD
- E. Tenir recursos limitats
- F. L'existència de criteris contraposats

7.2 Catalog

271. ¿Quina afirmació NO és certa respecte al catàleg?

- A. Conté la informació del sistema que el propi SGBD necessita per funcionar
- B. La seva estructura i continguts difereixen d'un SGBD a un altre, tot i que existeix un conjunt de vistes standard
- C. Conté totes les modificacions (insercions, modificacions i esborrats) realitzades des de l'última còpia de seguretat
- D. És útil per gestionar i afinar el funcionament de la base de dades

272. ¿Quin dels següents continguts del catàleg NO considerem estàtic?

- A. Informació de les vistes, com ara el seu nom, o la consulta associada

- B. Informació dels usuaris, com ara els intents de connexió
 - C. Informació de les taules, com ara el seu nom, atributs, o restriccions d'integritat
 - D. Informació dels índexs, com ara el seu nom, tipus, o atributs implicats
 - E. Paràmetres del sistema, com ara la grandària del *pool* de *buffers* o la grandària de pàgina
273. ¿Quin dels següents continguts del catàleg NO considerem dinàmic?
- A. Informació de les taules, com ara la seva cardinalitat o el nombre de blocs
 - B. Informació dels usuaris, com ara els intents de connexió
 - C. Informació de les vistes, com ara si estan materialitzades o es poden expandir
 - D. Informació dels índexs, com ara la seva alçada o el rang de valors
274. L'estàndard SQL'03 defineix quines han de ser les taules del catàleg de qualsevol SGBD.
- A. Cert
 - B. False
275. L'estàndard SQL'03 distingeix entre la informació del catàleg necessària per l'administrador i la necessària pels usuaris de la base dades.
- A. Cert
 - B. False

7.3 Access Structures

276. Les estructures d'accés (es a dir, els índexs) són ... respecte a les taules.
- A. Complementàries
 - B. Redundants
277. Un índex sempre té menys atributs que la taula.
- A. Cert
 - B. False
278. Sempre que posem un índex, la taula queda ordenada pels atributs indexats.
- A. Cert
 - B. False
279. ¿Què NO podem trobar en una entrada d'un índex?
- A. El registre sencer
 - B. Un mapa de bits
 - C. Una adreça física del registre
 - D. Una funció de *hash*
 - E. Una llista d'adreces físiques de registres
280. Els valors *null* sempre es troben a l'índex.
- A. Cert

- B. False
281. Els blocs de la taula contenen apuntadors als registres.
- A. Cert
B. False
282. Els blocs de la taula contenen metadades.
- A. Cert
B. False
283. Tots el registres continguts al mateix bloc tenen la mateixa longitud.
- A. Cert
B. False
284. Les entrades que trobem a les fulles d'un índex tipus arbre sempre estan ordenades.
- A. Cert
B. False
285. Tots els nodes d'un índex tipus arbre estan normalment plens al 100%.
- A. Cert
B. False
286. Un índex tipus *arbre* incrementa la grandària de la taula.
- A. Cert
B. False
287. Les entrades que trobem als *buckets* d'un índex tipus *hash* sempre estan ordenades.
- A. Cert
B. False
288. Tots els *buckets* d'un índex tipus *hash* estan normalment plens al 100%.
- A. Cert
B. False
289. Els *buckets* d'un índex tipus *hash* estan normalment més plens que els nodes d'un tipus arbre.
- A. Cert
B. False
290. Un índex tipus *hash* incrementa la grandària de la taula.
- A. Cert
B. False
291. Un índex tipus *cluster* manté ordenades les dades de la taula segons l'atribut d'indexació.
- A. Cert
B. False
292. Un índex tipus *cluster* incrementa la grandària de la taula.
- A. Cert
B. False

7.4 Size of Structures

293. $|T|$ representa el nombre de ... de la taula T .
- A. Atributs
 - B. *Bytes*
 - C. Files
 - D. Blocs
294. ¿Quin és típicament l'ordre (normalment representat per la lletra "d") d'un índex tipus arbre?
- A. Menor que 10
 - B. Entre 10 i 50
 - C. Entre 50 i 100
 - D. Major que 100
295. Els índexs ocupen sempre menys espai que la taula i conseqüentment mai generaran problemes d'espai.
- A. Cert
 - B. False
296. Un índex tipus arbre sempre ocupa menys espai que l'índex *cluster* corresponent.
- A. Cert
 - B. False
297. El nombre de blocs ocupats per qualsevol estructura de dades, sempre ha de ser un nombre enter.
- A. Cert
 - B. False

7.5 Access Costs

298. El cost d'utilitzar un índex sempre és més petit que el de llegir la taula sencera.
- A. Cert
 - B. False
299. Per estimar el cost d'accedir a una certa estructura, comptem només el cost corresponent a accedir els blocs de la taula.
- A. Cert
 - B. False
300. Per estimar el cost d'accedir a una certa estructura, comptem només el cost corresponent a accedir els blocs de disc.
- A. Cert
 - B. False
301. El cost d'accés estimat per a una estructura d'accés, sempre ha de ser un nombre enter.
- A. Cert
 - B. False

7.6 Index Choice

302. Els índexs tipus arbre i *hash* són molt útils quan l'atribut indexat té molts repetits.
- A. Cert
 - B. False
303. Posar un índex sempre és útil, independentment del tipus de consulta.
- A. Cert
 - B. False
304. Posar índexs mai empitjorà el temps de resposta del sistema.
- A. Cert
 - B. False
305. El millor és sempre definir tants índexs com puguem, però sense arribar a omplir el disc del tot.
- A. Cert
 - B. False
306. Els índexs tipus arbre i *hash* són molt útils en consultes amb condicions poc selectives.
- A. Cert
 - B. False
307. Els índexs tipus *hash* només són útils si tinc una condició de selecció per igualtat.
- A. Cert
 - B. False
308. Els índexs són útils a qualsevol taula.
- A. Cert
 - B. False
309. NO haig de definir un índex tipus arbre o *hash* si l'atribut té pocs valors.
- A. Cert
 - B. False
310. *Index-only query answering* vol dir que el SGBD no accedirà la taula corresponent per a resoldre la consulta.
- A. Cert
 - B. False

8 Query Optimization Phases: Semantic, Syntactic and Physical

8.1 Optimization Phases

311. ¿En quina posició se situa l'optimització dins del procés de processament de consultes que du a terme el gestor de consultes?
- A. Primera
 - B. Segona
 - C. Tercera
 - D. Última
312. L'optimització de consultes tradueix d'un llenguatge declaratiu a un de procedural.
- A. Cert
 - B. Fals
313. ¿Quina NO és una de les tres fases de l'optimització de consultes?
- A. Sintàctica
 - B. Semàntica
 - C. Física
 - D. Lògica
314. L'optimització escaneja les taules involucrades en la consulta per a calcular les estadístiques que necessita.
- A. Cert
 - B. Fals
315. L'optimitzador de consultes sempre troba el millor pla d'accés.
- A. Cert
 - B. Fals
316. L'optimitzador prioritza acabar ràpid la tasca d'optimització per davant de trobar el millor pla d'accés.
- A. Cert
 - B. Fals

8.2 Semantic Optimization

317. L'optimització semàntica canvia el llenguatge de representació de la consulta.
- A. Cert
 - B. Fals
318. L'optimització semàntica considera només la pròpia consulta i les lleis de la lògica.
- A. Cert

B. Fals

319. L'optimització semàntica pot fer que el cost d'execució d'una consulta sigui zero.

A. Cert

B. Fals

320. L'optimització semàntica pot fer més curta la clàusula *WHERE*.

A. Cert

B. Fals

321. L'optimització semàntica pot fer més llarga la clàusula *WHERE*.

A. Cert

B. Fals

8.3 Syntactic Optimization

322. L'optimització sintàctica canvia el llenguatge de representació de la consulta.

A. Cert

B. Fals

323. L'optimització sintàctica resol totes les vistes no-materialitzades que hi hagi a la consulta.

A. Cert

B. Fals

324. L'optimització sintàctica canvia el cost de la consulta.

A. Cert

B. Fals

325. Les dues regles heurístiques utilitzades a l'optimització sintàctica sempre milloren el cost de la consulta.

A. Cert

B. Fals

326. ¿Quina heurística utilitza l'optimització sintàctica?

A. Baixar les projeccions i seleccions tant com sigui possible

B. Reduir el nombre d'operacions tant com sigui possible

C. Pujar les projeccions i seleccions tant com sigui possible

D. Reduir el cost de la consulta tant com sigui possible

327. L'optimització sintàctica sempre redueix el nombre d'operacions a l'arbre sintàctic.

A. Cert

B. Fals

328. L'optimització sintàctica mai deixa una projecció just a sobre d'una fulla de l'arbre sintàctic.

A. Cert

- B. Fals
329. Després de l'optimització sintàctica poden quedar dos subgrafs iguals a l'arbre sintàctic.
- A. Cert
 - B. Fals
330. El resultat de l'optimització sintàctica sempre és un arbre.
- A. Cert
 - B. Fals
331. L'optimització sintàctica mai elimina operacions de l'arbre sintàctic.
- A. Cert
 - B. Fals

8.4 Equivalence Rules

332. Si una selecció té un predicat complex, sempre la podem dividir en dues seleccions consecutives.
- A. Cert
 - B. Fals
333. La selecció sempre commuta (sense posar ni treure res) amb la *join*.
- A. Cert
 - B. Fals
334. La selecció sempre commuta (sense posar ni treure res) amb qualsevol operació de conjunts (es a dir, unió, intersecció i diferència).
- A. Cert
 - B. Fals
335. La selecció sempre commuta (sense posar ni treure res) amb la projecció.
- A. Cert
 - B. Fals
336. La projecció sempre commuta (sense posar ni treure res) amb la *join*.
- A. Cert
 - B. Fals
337. La projecció sempre commuta (sense posar ni treure res) amb qualsevol operació de conjunts (es a dir, unió, intersecció i diferència).
- A. Cert
 - B. Fals
338. ¿Quines propietats compleix la *join* respecte a ella mateixa?
- A. Commutativa
 - B. Associativa
 - C. Cap de les dues
 - D. Totes dues

8.5 Physical Optimization

339. ¿Quin nom NO rep l'algorisme que segueix l'*execution manager* per obtenir el resultat d'una consulta?
- A. Pla d'accés de la consulta
 - B. Pla d'execució de la consulta
 - C. Arbre de procés de la consulta
 - D. Arbre d'execució de la consulta
340. ¿Què NO considera l'optimització física per a generar el pla d'execució d'una consulta?
- A. Les estructures físiques disponibles
 - B. Els camins d'accés que permet el predicat de la consulta
 - C. Els algorismes que té disponibles el SGBD
 - D. Les propietats ACID
341. ¿Quina operació deixa de ser explícita en l'arbre de procés?
- A. *Join*
 - B. Projecció
 - C. Unió
 - D. Selecció
342. L'arbre de procés mai té menys operacions que l'arbre sintàctic un cop optimitzat.
- A. Cert
 - B. Fals
343. L'arbre de procés mai té més operacions que l'arbre sintàctic un cop optimitzat.
- A. Cert
 - B. Fals
344. ¿Què NO genera alternatives en l'espai de cerca de la optimització basada en costos?
- A. Les estructures d'accés disponibles
 - B. Els algorismes existents per cada operació
 - C. L'ordre de les *joins*
 - D. La utilització de la *cache*
345. ¿Quines propietats de la *join* utilitza l'optimització física per a generar possibles alternatives d'execució?
- A. Associativitat i transitivitat
 - B. Commutativitat i transitivitat
 - C. Commutativitat i associativitat
 - D. Totes tres propietats
346. L'encarrilament (*pipelining*) és una tècnica d'execució de consultes que evita la materialització de resultats intermitjos.

- A. Cert
 - B. Fals
347. L'encarrilament (*pipelining*) es pot fer servir només quan tenim un esquema en estrella.
- A. Cert
 - B. Fals
348. Quan fem encarrilament (*pipelining*), l'ordre de les *joins* és ...
- A. Decreixent (de més selectiva a menys selectiva)
 - B. Creixent (de menys selectiva a més selectiva)
 - C. Irrellevant
 - D. El mateix que l'ordre dels atributs a la clau primària de la taula de fets
349. El cost d'una operació de l'arbre de procés és el cost de llegir l'entrada i d'executar la pròpia operació.
- A. Cert
 - B. Fals
350. El cost d'una consulta és la suma dels costos de totes les operacions del seu arbre de procés.
- A. Cert
 - B. Fals

9 Query Optimization Costs: Selection, Sorting and Projection

9.1 Intermediate Results

351. L'optimitzador calcula el cost de totes les alternatives d'execució que genera.
- A. Cert
 - B. Fals
352. El factor de selecció de qualsevol operació és el percentatge de files al resultat respecte a mínim de files que podem tenir.
- A. Cert
 - B. Fals
353. Un factor de selecció "1" és el més selectiu que pot haver.
- A. Cert
 - B. Fals
354. La cardinalitat màxima d'una *join* coincideix amb la del producte cartesià.
- A. Cert
 - B. Fals
355. El càlcul del factor de selecció de la *UNION* és el mateix que el de la *UNION ALL*.
- A. Cert
 - B. Fals
356. El càlcul del factor de selecció de la *join* és el mateix que el de la intersecció.
- A. Cert
 - B. Fals
357. Les cardinalitats dels resultats intermitjos es calculen *top-down* a l'arbre de procés.
- A. Cert
 - B. Fals
358. El SGBD manté sempre actualitzades les estadístiques de les taules.
- A. Cert
 - B. Fals
359. El SGBD típicament assumeix una distribució normal dels valors de cada atribut.
- A. Cert
 - B. Fals
360. El SGBD típicament assumeix la independència estadística de tots els atributs de les taules.
- A. Cert
 - B. Fals
361. El SGBD sempre calcula totes les estadístiques de la base de dades de cop.
- A. Cert

- B. Fals
362. El SGBD pot calcular les estadístiques de la base de dades només a partir d'un mostreig.
- A. Cert
- B. Fals
363. El factor de selecció d'un predicat mai pot ser "0".
- A. Cert
- B. Fals
364. El factor de selecció d'un predicat mai pot ser "1".
- A. Cert
- B. Fals
365. El factor de selecció d'una clàusula "IN" és exactament el mateix que el d'un predicat complex amb la disjunció de les igualtats del mateix atribut amb cadascun dels valors al conjunt de la "IN".
- A. Cert
- B. Fals
366. ¿Què NO afecta al factor de selecció d'una *join*?
- A. Que un dels atributs sigui clau primària
- B. Que un dels atributs accepti valor nulls
- C. Que un dels atributs sigui clau forana
- D. Que un dels atributs tingui un check
367. El factor de selecció de la θ -*join* amb "<>" és el mateix que el del producte cartesià.
- A. Cert
- B. Fals
368. La longitud d'un registre és la suma de les longituds dels seus atributs.
- A. Cert
- B. Fals
369. Estimem el nombre de registres a un bloc com la grandària del bloc dividit per la longitud del registre arrodonit per excés.
- A. Cert
- B. Fals
370. Estimem el nombre de blocs d'una taula com la cardinalitat de la taula dividida pel nombre de registres per bloc arrodonit per excés.
- A. Cert
- B. Fals

9.2 Selection

371. El primer pas per a processar una selecció d'un predicat complex és posar-ho en forma normal disjuntiva.

- A. Cert
- B. Fals

372. Si després de posar el predicat lògic d'una selecció en forma normal conjuntiva una de les condicions dins d'un parèntesi no permet utilitzar cap índex, llavors no podem utilitzar-ne cap tampoc a les altres condicions dins del parèntesi.

- A. Cert
- B. Fals

373. Si després de posar el predicat lògic d'una selecció en forma normal conjuntiva un dels parèntesi queda negat, simplement l'eliminem del procés.

- A. Cert
- B. Fals

374. La resolució de predicats de selecció amb operacions de llistes de RID, fa que no haguem de fer mai cap comprovació sobre les dades.

- A. Cert
- B. Fals

375. La resolució de predicats de selecció amb operacions de llistes de RID, fa que no haguem de fer mai un *table scan*.

- A. Cert
- B. Fals

376. Els index tipus arbre serveixen per avaluar clàusules lògiques amb qualsevol tipus de comparació.

- A. Cert
- B. Fals

377. Els index tipus *hash* serveixen per avaluar clàusules lògiques només amb comparacions per igualtat.

- A. Cert
- B. Fals

378. Un *bitmaps* guarda un bit per cada fila de la taula.

- A. Cert
- B. Fals

379. Un *bitmap* guarda una llista de bits per cada valor diferent de la taula.

- A. Cert
- B. Fals

380. Les operacions amb *bitmaps* són equivalents a les operacions de llistes de RIDs.

- A. Cert
- B. Fals

381. Els índex tipus *bitmaps* serveixen per avaluar clàusules lògiques amb qualsevol tipus de comparació.

- A. Cert
- B. Fals

382. Un índex tipus arbre multi-atribut no necessàriament utilitza més espai que si indexem només el primer dels seus atributs.

- A. Cert
- B. Fals

383. Un índex tipus arbre multi-atribut permet resoldre qualsevol selecció amb un predicat lògic que involucri tots els atributs indexats.

- A. Cert
- B. Fals

384. L'ordre dels atributs en un índex tipus arbre multi-atribut és irrellevant per a les consultes que es puguin fer amb ell.

- A. Cert
- B. Fals

9.3 Sorting

385. L'operació d'ordenació pot aparèixer en l'arbre de procés de consultes que NO tinguin ORDER BY.

- A. Cert
- B. Fals

386. Un índex tipus arbre es pot utilitzar per ordenar les dades.

- A. Cert
- B. Fals

387. Un índex tipus *hash* es pot utilitzar per ordenar les dades.

- A. Cert
- B. Fals

388. ¿Quin algorisme d'ordenació a memòria utilitza el *External Merge Sort*?

- A. *Bubble Sort*
- B. *Quick Sort*
- C. Cap
- D. És irrellevant

389. El *External Merge Sort* requereix d'una zona d'espai temporal per a realitzar l'ordenació, de l'ordre de la grandària de la pròpia taula.

- A. Cert
- B. Fals

390. El nombre de vegades que l'algorisme *External Merge Sort* ha de llegir i escriure la taula es logarítmic respecte a la grandària de la pròpia taula.

- A. Cert
- B. Fals

9.4 Projection

391. L'operació de projecció sense eliminació de repetits no té cap cost associat quan la consulta té alguna altra operació.

- A. Cert
- B. Fals

392. L'eliminació de repetits considera les mateixes alternatives d'algorismes que l'ordenació.

- A. Cert
- B. Fals

10 Query Optimization Costs: Join

10.1 Clustered Structure

393. Una estructura *cluster* incrementa l'espai requetis per les taules que l'ocupen.

- A. Cert
- B. False

394. Una estructura *cluster* incrementa el cost d'accedir les dades de només una de les taules que l'ocupen.

- A. Cert
- B. False

395. L'opció que dona el cost més baix per a fer una *join* es sempre tenir les dues taules en una estructura *cluster*.

- A. Cert
- B. Fals

10.2 Nested Loops

396. L'algorisme de *Row Nested Loops* és simètric.

- A. Cert
- B. Fals

397. L'algorisme de *Row Nested Loops* requereix l'existència d'un índex a priori.

- A. Cert
- B. Fals

398. L'algorisme de *Row Nested Loops* només es pot utilitzar si, en l'arbre de procés, l'operació de *join* està situada directament sobre la taula de l'índex utilitzat.

- A. Cert
- B. Fals

399. L'algorisme de *Row Nested Loops* només es pot utilitzar si la comparació de la *join* és la igualtat.

- A. Cert
- B. Fals

400. El cost de l'algorisme de *Row Nested Loops* és sempre més baix si no hem d'accedir atributs de la taula interna que no siguin els del propi índex utilitzat per l'algorisme.

- A. Cert
- B. Fals

401. L'algorisme de *Row Nested Loops* només permet utilitzar un índex tipus *cluster* si requerim atributs de la taula interna que no siguin els del propi índex.

- A. Cert

B. Fals

402. L'algorisme de *Block Nested Loops* sempre es pot utilitzar.

A. Cert

B. Fals

403. L'algorisme de *Block Nested Loops* és simètric.

A. Cert

B. Fals

404. L'algorisme de *Block Nested Loops* sempre té un cost més baix si posem la taula més gran al bucle extern.

A. Cert

B. Fals

405. L'algorisme de *Block Nested Loops* requereix que una de les taules càpigui a memòria.

A. Cert

B. Fals

10.3 Hash-Join

406. L'algorisme de *Hash-join* requereix l'existència d'un índex tipus *hash* a priori.

A. Cert

B. Fals

407. L'algorisme de *Hash-join* només es pot utilitzar si la comparació de la *join* és la igualtat.

A. Cert

B. Fals

408. L'algorisme de *Hash-join* és simètric.

A. Cert

B. Fals

409. L'algorisme de *Hash-join* requereix que una de les taules càpigui a memòria.

A. Cert

B. Fals

410. L'algorisme de *Hash-join* requereix dues passades si una de les taules no hi cap a memòria.

A. Cert

B. Fals

411. Si la taula més petita no hi cap a memòria, l'algorisme de *Hash-join* particiona les dues.

A. Cert

B. Fals

10.4 Sort-Match

412. L'algorisme de *Sort-Match* només es pot utilitzar si la comparació de la *join* és la igualtat.
- A. Cert
 - B. Fals
413. El cost de l'algorisme de *Sort-Match* depèn de la comparació de la *join*.
- A. Cert
 - B. Fals
414. L'algorisme de *Sort-Match* requereix que alguna de les taules estigui ordenada a priori.
- A. Cert
 - B. Fals
415. L'algorisme de *Sort-Match* requereix l'existència d'un índex *cluster* a priori.
- A. Cert
 - B. Fals
416. L'algorisme de *Sort-Match* deixa el resultat ordenat per l'atribut de *join*.
- A. Cert
 - B. Fals

11 Parametrization and Tuning

11.1 Tablespaces

417. ¿Quin NO és un dels tres espais d'un SGBD?
- A. Lògic
 - B. Físic
 - C. Conceptual
 - D. Virtual
418. ¿Què NO hi ha a l'espai lògic d'un SGBD?
- A. Files
 - B. Vistes
 - C. Columnnes
 - D. Taules
419. ¿Què NO hi ha a l'espai virtual d'un SGBD?
- A. Pàgines
 - B. *Clusters*
 - C. Particions
 - D. Índexs
 - E. Vistes
 - F. Taules
 - G. *Tablespaces*
420. ¿Què NO hi ha a l'espai físic d'un SGBD?
- A. Fitxers
 - B. Índexs
 - C. Blocks
 - D. Extensions
421. Les extensions serveixen per a garantir que el sistema operatiu assigni espai físicament consecutiu al disc.
- A. Cert
 - B. Fals
422. Un *Tablespace* pot tenir associats més d'un fitxer.
- A. Cert
 - B. Fals
423. Els *Tablespaces* faciliten disposar d'espai il·limitat a la base de dades.

- A. Cert
- B. Fals

424. Cal definir un *Tablespace* per a cada usuari del sistema.

- A. Cert
- B. Fals

425. Cal definir un *Tablespace* per a cada patró d'accés diferent al sistema.

- A. Cert
- B. Fals

11.2 Parameters

426. Els paràmetres del SGBD serveixen per a configurar el comportament dels seus subsistemes (com ara el gestor de consultes, el planificador, o el gestor de dades).

- A. Cert
- B. Fals

427. El *fillfactor* és el percentatge màxim que pot tenir mai ple un bloc de la taula.

- A. Cert
- B. Fals

428. Reduir el *fillfactor* sempre fa que les taules ocupin més espai.

- A. Cert
- B. Fals

429. Reduir el *fillfactor* fa que les modificacions de les dades sempre siguin més ràpides.

- A. Cert
- B. Fals

11.3 Bitmap Indexes

430. El nombre de repeticions de cada valor es irrellevant per a la utilitat d'un índex tipus arbre.

- A. Cert
- B. Fals

431. Un cop s'ha creat un índex tipus *bitmap*, es poden continuar fent insercions a la taula, però no de valors nous de l'atribut indexat.

- A. Cert
- B. Fals

432. Fer una inserció a una taula que té un índex tipus *bitmap* no incrementa la grandària de l'índex, tret que el valor de l'atribut corresponent no existís abans a la taula.

- A. Cert

B. Fals

433. Fer una modificació (*UPDATE*) a una taula que té un índex tipus *bitmap* no incrementa la grandària de l'índex, tret que el nou valor assignat a l'atribut corresponent no existís abans a la taula.

A. Cert

B. Fals

434. En el cas de l'índex *bitmap*, el factor de selecció indica el percentatge de blocs de la taula que caldrà accedir.

A. Cert

B. Fals

435. L'índex tipus *bitmap* és especialment útil en atributs *UNIQUE*.

A. Cert

B. Fals

436. ¿Quin tipus d'índex és millor en cas de consultes que involucrin múltiples valors d'un atribut?

A. *Hash*

B. *Bitmap*

C. *Arbre*

D. *Cluster*

437. ¿Quin tipus d'índex es millor en cas que l'atribut indexat tingui moltes repeticions?

A. *Bitmap*

B. *Cluster*

C. *Hash*

D. *Arbre*

438. L'índex tipus *bitmap* millora el temps de resposta d'una consulta quan el factor de selecció és inferior al 50%.

A. Cert

B. Fals

439. L'índex tipus *bitmap* incrementa el grau de concurrència del sistema.

A. Cert

B. Fals

440. L'índex tipus *bitmap* no es pot utilitzar si l'atribut indexat conté valors *null*.

A. Cert

B. Fals

441. L'índex tipus *bitmap* facilita la comprovació d'unicitat en l'atribut corresponent.

A. Cert

B. Fals

11.4 Tuning

442. ¿Quin tipus d'usuari s'encarrega del *tuning* de la base de dades?
- A. Dissenyador
 - B. Programador d'aplicacions
 - C. Administrador
 - D. Tots els anteriors
443. ¿Què NO forma part de l'entrada del procés de millora del rendiment del sistema?
- A. Llista d'operacions de modificació, juntament amb les seves freqüències
 - B. Espai de disc disponible
 - C. Llista d'operacions de consulta, juntament amb les seves freqüències
 - D. Llista de restriccions d'integritat, juntament amb les probabilitats de ser violades
 - E. Objectiu de rendiment a assolir
444. ¿Què NO forma part de la sortida del procés de millora del rendiment del sistema?
- A. Rescriptura de cada consulta (segons els criteris de l'optimització semàntica)
 - B. Normalització/Desnormalització de les taules
 - C. Conjunt de vistes materialitzades
 - D. Conjunt d'índexs
 - E. Particionament de les taules
445. ¿Quina dada de les que no trobem al pla d'accés NO es rellevant per a fer *tuning* de l'execució de les consultes?
- A. Nombre d'abraçades mortals (*deadlocks*)
 - B. Temps en les cues de bloquejos
 - C. Grandària del dietari (*log*)
 - D. Nombre de bloquejos
 - E. *Hits* a la *cache*
446. El nombre d'índexs que podem crear a una base de dades es lineal respecte al nombre de taules i atributs existents.
- A. Cert
 - B. Fals
447. L'única limitació que tenim per a crear índexs és l'espai de disc disponible.
- A. Cert
 - B. Fals
448. Un índex que no sigui de tipus *cluster* mai empitjorarà l'estimació del temps d'execució d'una consulta.

- A. Cert
 - B. Fals
449. És millor no posar índexos a les taules petites.
- A. Cert
 - B. Fals
450. Pot ser que un índex no millori el rendiment de cap operació DML concreta, però tot i així sigui bo crear-ho des del punt de vista del rendiment del sistema.
- A. Cert
 - B. Fals
451. El predicat de les consultes es irrellevant per a triar el tipus d'índex d'una taula.
- A. Cert
 - B. Fals
452. L'ordre dels atributs d'un índex multiatribut afecta a la seva utilitat.
- A. Cert
 - B. Fals
453. Una taula pot tenir com a màxim un índex tipus *cluster*.
- A. Cert
 - B. Fals
454. Com més repeticions tingui un atribut, millor serà posar un índex tipus *hash* en comptes d'un arbre.
- A. Cert
 - B. Fals
455. Com més repeticions tingui un atribut, millor serà posar un índex tipus *bitmap* en comptes d'un arbre.
- A. Cert
 - B. Fals
456. L'algorisme *greedy* de selecció d'índexs és bàsicament el mateix que el de selecció de vistes materialitzades.
- A. Cert
 - B. Fals

11.5 Example of Index Selection

457. Quan utilitzem l'algorisme *greedy*, cal calcular el temps d'execució considerant també les estructures que no hi càpiguen, perquè podem guanyar l'espai necessari per elles eliminant una altra estructura que haguem triat amb anterioritat.
- A. Cert
 - B. Fals

12 Transactions

12.1 Isolation

458. Pel que fa a l'aïllament, volem tenir només històries serials.
- A. Cert
 - B. Fals
459. El nivell d'aïllament *read uncommitted* només bloqueja escriptures.
- A. Cert
 - B. Fals
460. El nivell d'aïllament *read committed* manté tots els bloquejos fins al final de la transacció.
- A. Cert
 - B. Fals
461. El nivell d'aïllament *repeatable read* genera dos bloquejos per cada lectura.
- A. Cert
 - B. Fals
462. El nivell d'aïllament *serializable* sempre bloqueja tota la taula fins al final de la transacció.
- A. Cert
 - B. Fals
463. Com més garanties d'aïllament tinguem, millor serà el rendiment del sistema.
- A. Cert
 - B. Fals
464. ¿Quin dels següents conceptes NO afecta al temps d'execució de la transacció?
- A. Tipus de bloquejos
 - B. Moment d'assignació del timestamp
 - C. Nombre de bloquejos
 - D. Moment d'alliberament dels bloquejos
-
465. ¿Quina de les següents accions NO serveix per millorar el rendiment de les transaccions?
- A. Trossejar les transaccions
 - B. Assignar el timestamp el més tard possible
 - C. Relaxar el nivell d'aïllament
 - D. Eliminar bloquejos innecessaris
 - E. Configurar l'interval de detecció de *deadlock* adequat
 - F. Evitar (o endarrerir) l'accés a grànuls molt demandats
 - G. Utilitzar les sentències de DDL quan hi hagi pocs usuaris

12.2 Multi-version Concurrency Control

466. El control de concurrència multi-versió fa que dues transaccions que s'executin concurrentment puguin llegir dades diferents.

- A. Cert
- B. Fals

467. Amb control de concurrència multi-versió, cada operació d'escriptura genera una nova versió del grànul.

- A. Cert
- B. Fals

468. Amb control de concurrència multi-versió, l'usuari ha de decidir quina versió del grànul llegeix.

- A. Cert
- B. Fals

469. Amb control de concurrència multi-versió, les operacions de lectura no bloquegen mai res.

- A. Cert
- B. Fals

470. Amb control de concurrència multi-versió, només hi ha bloquejos entre escriptures.

- A. Cert
- B. Fals

471. Amb control de concurrència multi-versió, fer *rollback* d'una transacció és equivalent a eliminar les versions generades per la transacció.

- A. Cert
- B. Fals

472. Amb control de concurrència multi-versió, tenim implícit una base de dades temporal amb temps de transacció.

- A. Cert
- B. Fals

473. El principal problema del control de concurrència multi-versió, és que hem d'esborrar les versions obsoletes dels grànuls per a mantenir la grandaria de la base de dades dins d'uns límits.

- A. Cert
- B. Fals

474. Fent *vacuum*, eliminem les versions obsoletes de les dades en el control de concurrència multi-versió.

- A. Cert
- B. Fals

475. Amb un control de concurrència multi-versió basat en bloquejos, s'aplica un control de concurrència diferent depenent de si la transacció és *read-only* o *read-write*.

- A. Cert
- B. Fals

476. Amb un control de concurrència multi-versió basat en bloquejos, les transaccions *read-only* assignen un *timestamp* ...

- A. En acabar la transacció
- B. A l'inici de la transacció
- C. No n'assignen mai cap
- D. En intentar accedir un grànul bloquejat

477. Amb un control de concurrència multi-versió basat en bloquejos, les transaccions *read-write* assignen un *timestamp* ...

- A. En intentar accedir un grànul bloquejat
- B. En acabar la transacció
- C. A l'inici de la transacció
- D. No n'assignen mai cap
- E. Al fer la primera operació d'escriptura

12.3 Atomicity and Durability

478. ¿Quina NO és una raó per a necessitar una reconstrucció?

- A. Incendi
- B. Fallada del disc
- C. Fallada elèctrica
- D. Totes les anteriors ho són

479. ¿Quina de les quatre propietats ACID està relacionada principalment amb la restauració?

- A. Atomicitat
- B. Consistència
- C. aïllament
- D. Durabilitat

480. ¿Quina de les quatre propietats ACID està relacionada principalment amb la reconstrucció?

- A. Atomicitat
- B. Consistència
- C. aïllament
- D. Durabilitat

481. ¿Què fa el gestor de la cache quan rep una petició de *Fetch*?

- A. Llegeix de la memòria i escriu al disc
- B. Llegeix del disc i la memòria

- C. Llegeix del disc i escriu a la memòria
 - D. Escriu tant al disc com a la memòria
482. ¿Què fa el gestor de la cache quan rep una petició de *Flush*?
- A. Escriu tant al disc com a la memòria
 - B. Llegeix del disc i la memòria
 - C. Llegeix de la memòria i escriu al disc
 - D. Llegeix del disc i escriu a la memòria
483. ¿Què NO es guarda al dietari?
- A. Tipus d'operació
 - B. Pla d'accés de la consulta
 - C. Punter a l'operació anterior de la transacció
 - D. Valor vell del grànul modificat
 - E. Identificador de l'objecte
 - F. Punter a l'operació següent de la transacció
 - G. Valor nou del grànul modificat
484. ¿Quan NO es fa un *flush* dels buffers del SGBD?
- A. Durant un backup
 - B. Quan el nombre de pàgines de memòria modificades supera un cert llindar
 - C. Quan el dietari s'omple
 - D. A intervals regulars
 - E. Durant una operació de DDL
485. El *Write Ahead Log Protocol* estableix que abans de confirmar qualsevol operació de DML, aquesta s'ha d'escriure al dietari (*log*).
- A. Cert
 - B. Fals
486. ¿Com evita el SGBD que s'ompli el dietari (*log*)?
- A. Dedicant-li un disc de forma exclusiva
 - B. Limitant el nombre d'escriptures al mínim indispensable
 - C. Esborrant-ho i creant-ho de nou a intervals regulars
 - D. Gestionant el fitxer com si fos cíclic
487. Si fem *backups* de forma regular, no ens cal el dietari (*log*) per a reconstruir la base de dades.
- A. Cert
 - B. Fals
488. Just quan acabem de fer un *backup*, podem esborrant el dietari (*log*).

- A. Cert
 - B. Fals
489. Si configurem de forma adequada el dietari (*log*), no ens cal fer *backups*.
- A. Cert
 - B. Fals
490. Mentre s'executa un *backup*, el rendiment del sistema empitjora.
- A. Cert
 - B. Fals
491. És important posar el dietari (*log*) en un disc dedicat exclusivament per a ell, perquè així evitem moviments innecessaris del capçal i afavorim la seva escriptura seqüencial.
- A. Cert
 - B. Fals
492. Endarrerir els *flush* tan com sigui possible millora el rendiment del sistema.
- A. Cert
 - B. Fals

12.4 Transaction Chopping

493. ¿Quin NO és un problema provocat per que les transaccions *read-write* siguin innecessàriament llargues?
- A. És més probable que el sistema falli durant la seva execució
 - B. La pròpia transacció és probable que es bloquegi
 - C. Altres transaccions han d'esperar massa
 - D. En cas de fallida, la transacció trigarà molt a recuperar-se
 - E. Totes les anteriors ho són
494. Trossejar les transaccions *read-write* millora el rendiment del sistema.
- A. Cert
 - B. Fals
495. El fet que trossejar una transacció afecti o no al seu aïllament depèn només d'ella mateixa i no de cap altra transacció que es pugui executar de forma concurrent.
- A. Cert
 - B. Fals
496. Les transaccions llargues són un problema tant pel control de concurrència, com per a la recuperació.

A. Cert

B. Fals

497. Si concatenem (*chain*) dues transaccions, no s'alliberen els bloquejos de la primera fins que acaba la segona.

A. Cert

B. Fals

498. Si concatenem (*chain*) dues transaccions, no s'alliberen els recursos (p.e., memoria) de la primera i aquests es reutilitzen en la segona, que manté la mateixa configuració (p.e., nivell d'aïllament).

A. Cert

B. Fals

12.5 Alternative Transaction Manager Architectures

499. Si totes les transaccions són prou curtes, no cal paral·lelitzar les escriptures i podem substituir el control de concurrència per un sistema de cues.

A. Cert

B. Fals

500. Si tenim prou memòria com per a contenir tota la base de dades, podem fer directament allà totes les operacions i no cal escriure mai al disc.

A. Cert

B. Fals

501. Si tenim un sistema que només fa lectures, podem prescindir tant dels mecanismes de control de concurrència com dels de recuperació.

A. Cert

B. Fals

A Respostes

Pregunta	Resposta	Pregunta	Resposta	Pregunta	Resposta	Pregunta	Resposta
1	B	45	D	89	A	133	A
2	B	46	A	90	B	134	B
3	A	47	B	91	B	135	D
4	D	48	A	92	A	136	B
5	D	49	B	93	B	137	A
6	C	50	D	94	B	138	B
7	B	51	B	95	B	139	A
8	B	52	A	96	C	140	B
9	B	53	B	97	D	141	B
10	B	54	D	98	B	142	E
11	B	55	B	99	A	143	A
12	A	56	B	100	A	144	A
13	A	57	B	101	A	145	B
14	A	58	B	102	B	146	B
15	A	59	B	103	A	147	A
16	C	60	D	104	B	148	A
17	A	61	A	105	A	149	B
18	D	62	A	106	C	150	C
19	B	63	A	107	A	151	B
20	A	64	D	108	D	152	B
21	A	65	D	109	A	153	C
22	B	66	B	110	C	154	A
23	B	67	A	111	D	155	B
24	C	68	D	112	A	156	A
25	B	69	A	113	B	157	B
26	A	70	A	114	B	158	C
27	B	71	B	115	A	159	A
28	B	72	A	116	D	160	B
29	A	73	A	117	A	161	C
30	D	74	C	118	B	162	A
31	C	75	D	119	B	163	B
32	A	76	A	120	A	164	A
33	A	77	A	121	B	165	B
34	D	78	C	122	B	166	A
35	A	79	A	123	A	167	B
36	B	80	B	124	B	168	A
37	B	81	C	125	A	169	B
38	B	82	D	126	A	170	E
39	B	83	B	127	B	171	A
40	B	84	B	128	A	172	A
41	C	85	B	129	A	173	A
42	B	86	A	130	B	174	C
43	D	87	D	131	B	175	B
44	A	88	B	132	B	176	A

Pregunta	Resposta	Pregunta	Resposta	Pregunta	Resposta	Pregunta	Resposta
177	B	221	A	265	B	309	A
178	C	222	C	266	C	310	A
179	B	223	C	267	A	311	D
180	B	224	A	268	A	312	A
181	A	225	A	269	A	313	D
182	A	226	C	270	C	314	B
183	C	227	E	271	C	315	B
184	C	228	C	272	B	316	A
185	B	229	B	273	C	317	B
186	A	230	A	274	B	318	B
187	C	231	B	275	A	319	A
188	C	232	C	276	A	320	A
189	A	233	D	277	B	321	A
190	A	234	D	278	B	322	A
191	B	235	A	279	D	323	A
192	A	236	A	280	B	324	A
193	B	237	A	281	A	325	B
194	A	238	B	282	A	326	A
195	A	239	B	283	B	327	B
196	D	240	A	284	A	328	B
197	A	241	A	285	B	329	B
198	A	242	B	286	B	330	B
199	A	243	B	287	B	331	B
200	A	244	B	288	B	332	B
201	B	245	A	289	A	333	A
202	B	246	E	290	B	334	A
203	A	247	A	291	A	335	B
204	B	248	A	292	A	336	B
205	B	249	B	293	C	337	B
206	A	250	A	294	D	338	D
207	A	251	A	295	B	339	D
208	A	252	B	296	A	340	D
209	A	253	B	297	A	341	B
210	A	254	B	298	B	342	B
211	A	255	A	299	B	343	B
212	B	256	E	300	A	344	D
213	A	257	D	301	B	345	C
214	A	258	A	302	B	346	A
215	B	259	C	303	B	347	B
216	A	260	A	304	B	348	A
217	A	261	E	305	B	349	B
218	B	262	B	306	B	350	A
219	B	263	B	307	A	351	A
220	A	264	B	308	B	352	B

Pregunta	Resposta	Pregunta	Resposta	Pregunta	Resposta	Pregunta	Resposta
353	B	397	A	441	B	485	B
354	A	398	A	442	D	486	D
355	B	399	B	443	D	487	B
356	A	400	A	444	A	488	A
357	B	401	B	445	C	489	B
358	B	402	A	446	B	490	A
359	B	403	B	447	B	491	A
360	A	404	B	448	A	492	A
361	B	405	B	449	A	493	E
362	A	406	B	450	B	494	A
363	B	407	A	451	B	495	B
364	B	408	B	452	A	496	A
365	A	409	B	453	A	497	B
366	D	410	B	454	B	498	A
367	A	411	A	455	A	499	A
368	B	412	B	456	A	500	B
369	B	413	A	457	B	501	A
370	A	414	B	458	B		
371	B	415	B	459	A		
372	A	416	A	460	B		
373	B	417	C	461	B		
374	B	418	B	462	A		
375	B	419	F	463	B		
376	B	420	B	464	B		
377	A	421	A	465	B		
378	B	422	A	466	A		
379	A	423	A	467	A		
380	A	424	B	468	B		
381	B	425	A	469	A		
382	B	426	A	470	A		
383	B	427	B	471	A		
384	B	428	A	472	A		
385	A	429	B	473	A		
386	A	430	B	474	A		
387	B	431	B	475	A		
388	D	432	B	476	B		
389	A	433	A	477	B		
390	A	434	B	478	C		
391	A	435	B	479	A		
392	B	436	B	480	D		
393	A	437	A	481	C		
394	A	438	B	482	C		
395	B	439	B	483	B		
396	B	440	B	484	E		