

# Examen S01a Historia

## Calificación del examen = 9 (de 9)

1.- (1 punto)

El primer medio de almacenamiento masivo de datos aplicado a la computación electrónica es

- ☐ el disco magnético
- ☐ el disco flexible
- ☒ la tarjeta perforada.
- ☐ la cassette

Obtenido = 1

2.- (1 punto)

Ordenados de más a menos capacidad de almacenamiento, discos flexibles de tamaño aproximado de

- ☐ 8, 5 y 3 Mbytes

- ☒ 3, 5 y 8 pulgadas
- ☐ 8, 5, y 3 pulgadas.
- ☐ 5, 3 y 8 pulgadas

Obtenido = 1

3.- (1 punto)

La tecnología magnética aplicada al almacenamiento masivo de datos se aplica a

- ☐ salvo a las tarjetas perforadas, a todos
- ☐ discos duros, discos flexibles y discos flash
- ☒ discos duros y flexibles
- ☐ CD, DVD y Blu Ray

Obtenido = 1

4.- (1 punto)

Antes de la llegada de la tecnología basada en bases de datos, la definición del fichero estaba incluida en el código de los programas por lo que un cambio de un dato en todos los programas que usan ese fichero obligan a

- ☒ recompilar
- ☒ recodificar
- ☒ reenlazar

☐ reversionar

Obtenido = 1

5.- (1 punto)

Los problemas detectados en los sistemas de ficheros antes de la llegada de las técnicas de bases de datos se pueden resumir en

☐ los programas dependen de los datos

☒ los programas dependen de datos que no usan

☒ seguridad insuficiente

☒ redundancia de datos

Obtenido = 1

6.- (1 punto)

Hablando del almacenamiento persistente de datos en computación, por seguridad entendemos

☐ dependencia de los programas de datos que no usan

☒ autorizaciones

☒ acceso simultáneo a los mismos datos

☒ recuperación ante desastres

Obtenido = 1

7.- (1 punto)

La crisis del software, entre otras, tenía las siguientes características

- ☐ los proyectos eran demasiado pequeños
- ☐ los programadores no trabajaban lo suficiente
- ☒ el software era ineficiente
- ☒ el software a menudo no cumplía con los requerimientos

Obtenido = 1

8.- (1 punto)

La crisis del software, entre otras, tenía las siguientes características

- ☐ los proyectos no se explicaban bien
- ☐ el software se programaba en Cobol
- ☒ los proyectos se hacían inmanejables y el código difícil de mantener.
- ☒ el software tenía baja calidad

Obtenido = 1

9.- (1 punto)

La crisis del software, entre otras, tenía las siguientes características

- ☒ el presupuesto inicial se superaba con creces

☐ los proyectos no eran trasladables o traducibles a código máquina.

☒ los proyectos se terminaban fuera de plazo

☐ la ingeniería del software no servía

Obtenido = 1

Enviar

[Volver](#)

# Examen S01b Ficheros

**Calificación del examen = 21 (de 21)**

1.- (1 punto)

Por persistencia entendemos

- ☐ que el disco duro no se apaga nunca, en todo caso se deja en suspensión
- ☐ el almacenamiento conceptual
- ☒ apagar el ordenador y que el día siguiente pueda recuperar los datos almacenados ayer
- ☒ almacenar la información para su posterior recuperación y proceso

**Obtenido = 1**

2.- (1 punto)

La persistencia de datos

- ☒ se soluciona con ficheros

- ☐ no tiene nada que ver con los ficheros electrónicos
- ☐ no afecta a la recuperación y proceso de esos datos.
- ☒ se soluciona en la parte física de una base de datos

Obtenido = 1

3.- (1 punto)

La persistencia de datos se soporta con

- ☐ almacenamiento primario
- ☒ discos magnéticos y ópticos
- ☒ tarjetas perforadas y cintas magnéticas
- ☐ memoria RAM

Obtenido = 1

4.- (1 punto)

La persistencia se relaciona con

- ☐ el almacenamiento cuaternario
- ☐ el almacenamiento primario
- ☒ el almacenamiento secundario
- ☒ el almacenamiento terciario

Obtenido = 1

5.- (1 punto)

Por almacenamiento terciario o fuera de línea entendemos

- ☐ las bases de datos
- ☒ los discos ópticos y SSD
- ☒ los discos magnéticos y cintas
- ☐ los ficheros

Obtenido = 1

6.- (1 punto)

Son dispositivos secuenciales

- ☐ el disco magnético
- ☐ el disco flash
- ☒ la cinta
- ☒ la tarjeta perforada

Obtenido = 1

7.- (1 punto)

Son dispositivos aleatorios



- ☒ el disco flash
- ☐ la cinta
- ☐ la tarjeta perforada
- ☒ el disco magnético

Obtenido = 1

8.- (1 punto)

En el disco magnético son importantes los tiempos de

- ☒ transferencia de bloque
- ☒ búsqueda
- ☒ latencia
- ☐ envío

Obtenido = 1

9.- (1 punto)

El direccionamiento dentro de un disco duro incluye

- ☒ cilindro
- ☒ cabezal
- ☒ sector

☐ byte

Obtenido = 1

10.- (1 punto)

Los registros pueden tener longitud

☐ escasa

☒ mixta

☒ fija

☒ variable

Obtenido = 1

11.- (1 punto)

Los registros de longitud variable

☒ utilizan marcas para señalar el inicio y fin de los campos

☐ no son recomendables si el objetivo es una recuperación de datos fiable

☐ gastan más espacio que los de longitud fija por culpa de las marcas de campo.

☒ optimizan el espacio

Obtenido = 1

12.- (1 punto)

## Los registros de longitud fija

- ☐ nunca debería utilizarse con la tecnología actual
- ☒ facilitan la recuperación de registros concretos
- ☐ también necesitan marcas para identificar el final del registro.
- ☒ rellenan el espacio vacío con blancos

Obtenido = 1

13.- (1 punto)

Si en un fichero con organización *hash* hemos reservado espacio para 13 registros y utilizamos la fórmula  $h = n \bmod 13 + 1$  para posicionar cualquier nuevo registro, siendo  $n$  la clave de indexación y de tipo entero

- ☒ los registros identificados por 5 y 18 colisionarán en la misma posición.
- ☐ si inserto, en este orden, el registro con clave 5 y el 18, el 5 ocupará la primera posición física del fichero y el 18 la segunda.
- ☒ si ya tengo un registro con clave 5, insertar el 18 provocará el uso del área de desbormiento.

Obtenido = 1

14.- (1 punto)

En las operaciones con datos en almacenamiento secundario

- ☐ Mac OS y Sun solo utilizan almacenamiento primario
- ☒ aunque se trabaje con un registro, el disco se procesa por bloques o páginas

☐ si quiero recuperar un único byte, el sistema operativo no me proporciona herramienta alguna, tengo que especificar dentro del programa el cabezal, cilindro y sector del disco.

☒ antes hay que transferir la información desde el almacenamiento secundario al primario.

Obtenido = 1

15.- (1 punto)

La organización secuencial ordenada

☐ optimiza el espacio

☐ es la peor para obtener un listado ordenado por el campo clave.

☐ es la mejor para obtener un listado ordenado por cualquier campo.

☒ obliga a una reorganización del fichero cada vez que quiero insertar un registro intermedio.

Obtenido = 1

16.- (1 punto)

La organización secuencial pura

☐ es la mejor para obtener un listado ordenado por cualquier campo.

☐ optimiza la búsqueda de registros concretos

☐ obliga a una reorganización del fichero cada vez que quiero insertar un registro intermedio.

☒ es la mejor para añadir nuevos registros.

Obtenido = 1

17.- (1 punto)

Mis datos requieren actualizaciones, borrados e inserciones frecuentes, y listados ordenados por cualquier criterio, y la cantidad de registros puede variar enormemente, por lo tanto elegiré una organización

☐ secuencial ordenada

☐ hash

☒ indexada

☐ secuencial

Obtenido = 1

18.- (1 punto)

La organización aleatoria calculada (hash)

☐ los registros están, siempre, ordenados físicamente por un campo clave

☐ necesitan ficheros auxiliares para acceder mediante el valor clave y , después, direccionar al fichero principal de datos.

☒ pueden necesitar registros o áreas de desbordamiento

☐ me optimiza, en tiempo, los listados ordenados por un campo cualquiera

Obtenido = 1

19.- (1 punto)

Un índice denso

- ☐ no pueden ser multinivel
- ☒ requiere que el fichero de datos esté ordenado por un campo clave
- ☐ estructuran los registros en forma de árbol
- ☐ es un índice con muchos datos

Obtenido = 1

20.- (1 punto)

La organización indexada se basa en la idea de que

- ☐ la selección de un campo cualquiera para la ordenación de listados de datos es necesaria solo en muy contados casos.
- ☒ es más fácil manejar archivos pequeños que incluso podrían caber en RAM para encontrar un registro concreto
- ☐ no me preocupa el espacio ocupado en el disco.
- ☐ no se necesita insertar con demasiada frecuencia

Obtenido = 1

21.- (1 punto)

La búsqueda binaria

- ☐ solo se aplica a ficheros aleatorios

☐ es eficiente con ficheros desordenados

☒ es uno de los algoritmos habituales de búsqueda

☐ solo se aplica a los ficheros de datos, no a los de índice, si esta fuera la organización del fichero.

Obtenido = 1

Enviar

[Volver](#)

# Examen T2 modelos de datos

## Calificación del examen = 9 (de 9)

1.- (1 punto)

El propósito de los modelos de datos es

- ☒ proporcionar las herramientas necesarias para modelar un sistema de información.
- ☒ permitir obtener un esquema como representación de un sistema eliminando detalles irrelevantes.
- ☐ definir un lenguaje de programación como C, PHP o Java.
- ☐ introducir la definición de los ficheros dentro del código de programa.

Obtenido = 1

2.- (1 punto)

Un modelo de datos es

- ☒ casi lo podemos resumir en que es un lenguaje.



- ☐ Una fase más del desarrollo del software como el análisis, el diseño y la implementación
- ☐ un esquema conceptual.
- ☒ la herramienta intelectual que nos permite estructurar los datos de forma que se capte la semántica de los mismos.

Obtenido = 1

3.- (1 punto)

El esquema es

- ☒ una descripción de un sistema concreto.
- ☐ un lenguaje de programación como C, PHP o Java.
- ☒ el resultado de aplicar un determinado modelo de datos.
- ☐ una descripción de ficheros.

Obtenido = 1

4.- (1 punto)

En el ciclo de desarrollo de software

- ☒ hay seis fases: análisis, esquema conceptual, diseño, esquema lógico, implementación y esquema físico.
- ☒ el análisis es la primera fase.
- ☐ la implementación es el primer paso.

☒ es habitual generar tres esquemas: conceptual, lógico y físico.

Obtenido = 1

5.- (1 punto)

En cuanto a los esquemas resultado de las distintas fases de desarrollo de software

- ☐ ninguno depende de otro, el lógico no depende del conceptual, el físico no depende del conceptual, etc.
- ☐ el esquema conceptual se genera teniendo en cuenta que se va trabajar en una base de datos.
- ☒ el esquema físico es lo más cercano al disco duro, al mantenimiento de ficheros.
- ☒ el esquema conceptual NO depende de una máquina y software concreta.

Obtenido = 1

6.- (1 punto)

El lenguaje de definición de datos

- ☒ me permite construir un esquema de, por ejemplo, una base de datos relacional.
- ☐ puede ser navegacional o de especificación.
- ☐ tiene como algunos de sus cometidos principales permitir la inserción y modificación de registros.
- ☒ es parte de la definición del modelo de datos.

Obtenido = 1

7.- (1 punto)

Un modelo de datos es

- ☒ una teoría de especificación describiendo como una base de datos se estructura y se usa.
- ☐ un esquema de base datos como, por ejemplo, el de TiendaOnLine.
- ☐ todo aquello que se pueda consultar con una orden select.
- ☐ una base de datos ejemplo para todas las demás.

Obtenido = 1

8.- (1 punto)

UML

- ☐ significa Lenguaje de Múltiples Usos.
- ☐ es el modelo de datos usado para describir las tablas de las prácticas de la asignatura.
- ☒ es un lenguaje de modelado estandarizado y de propósito general en la ingeniería del software.
- ☐ es un lenguaje textual, NO gráfico.

Obtenido = 1

9.- (1 punto)

Los sistemas de información

- ☐ describen los datos conceptualmente.
- ☐ son modelos de datos.

- tienen como principal función manejar grandes volúmenes de datos tanto estructurados como no estructurados.
- generan lenguajes de definición y manipulación de datos.

Obtenido = 1

Enviar

[Volver](#)

# Examen T2B E-R

**Calificación del examen = 12 (de 12)**

1.- (1 punto)

El entidad-relación

- ☒ es un lenguaje gráfico.
- ☐ es un lenguaje textual, NO gráfico.
- ☐ es un modelo de datos de propósito particular.
- ☒ es un modelo de datos semántico.

**Obtenido = 1**

2.- (1 punto)

El entidad-relación

- ☒ es semántico y el modelo relacional es clásico.

- ☐ es el modelo de datos subyacente en el motor de MySQL, Oracle y SQL Server.
- ☐ genera esquemas físicos.
- ☒ tiene una relación estrecha con el modelo relacional

Obtenido = 1

3.- (1 punto)

El entidad-relación

- ☒ es semántico y gráfico.
- ☐ define ficheros, registros y campos.
- ☐ no sirve para describir otra cosa que no sea una base de datos.
- ☒ no tiene sistemas de gestión de bases de datos basados en él.

Obtenido = 1

4.- (1 punto)

Según este esquema E-R

□

- ☒ los usuarios siempre tienen email, dni, nombre y apellidos.
- ☐ todos los usuarios deben haber pedido algo.
- ☒ los usuarios pueden pedir todos los artículos existentes, si quieren.
- ☐ los usuarios solo pueden pedir N artículos, donde N es un valor indeterminado entre 0 y 1.

Obtenido = 1

5.- (1 punto)

Según este esquema E-R

□

- ☐ imagen es un atributo multivaluado.
- ☒ los artículos pueden no tener marca de fabricante.
- ☒ muchos artículos puede ser fabricados por la misma marca.
- ☐ todos los artículos tienen marca de fabricante.

Obtenido = 1

6.- (1 punto)

Según este esquema E-R

□

- ☐ los usuarios pueden tener varias localidades de residencia, la principal y la de veraneo, por ejemplo.
- ☐ los usuarios se identifican por su DNI.
- ☒ los usuarios siempre viven en al menos una localidad.
- ☒ las localidades pueden NO tener habitantes.

Obtenido = 1

7.- (1 punto)

En E-R, la generalización

- ☒ permite que los objetos especializados mantengan atributos propios y exclusivos.
- ☐ los objetos especializados no tienen identificador y, por lo tanto, puede haber duplicados.
- ☐ siempre tiene que tener al menos 2 clases de objetos especializadas.
- ☒ representa una relación de herencia, X es-un Y.

Obtenido = 1

8.- (1 punto)

En E-R, la agregación

- ☐ no admite atributos multivaluados.
- ☒ no es más que un caso particular de la agregación general de cualquier modelo de datos.
- ☐ consiste en encontrar subtipos dentro de una clase de objetos general.
- ☒ permite relacionar entidades con otra relación.

Obtenido = 1

9.- (1 punto)

En E-R, la generalización

- ☒ es una representación de subtipos de una clase de objetos general.
- ☐ puede ser al mismo tiempo parcial, total, disjunta y solpada.
- ☒ se define siempre con propiedades de cobertura.



☐ si es parcial no es disjunta. y si es total no es solapada.

Obtenido = 1

10.- (1 punto)

Son notaciones distintas del E-R

☐ por entidades.

☐ por ficheros.

☐ por tablas.

☒ Crow's foot

Obtenido = 1

11.- (1 punto)

En cuanto a las distintas notaciones del E-R

☐ no es cierto, solo hay una única notación.

☒ nos da igual, podremos obtener un esquema equivalente en cada una de ellas.

☐ nos referimos al nombre que le daremos al esquema de base de datos.

☐ unas sirven para el esquema conceptual, otras para el lógico y otras para el físico.

Obtenido = 1

12.- (1 punto)

Son notaciones distintas del E-R

☒ UML

☐ entidad, atributo y relación

☐ conceptual, lógica y física.

☐ jerárquica, en red y relacional.

Obtenido = 1

Enviar

[Volver](#)

# Examen T3 MR A

**Calificación del examen = 17 (de 17)**

1.- (1 punto)

El modelo relacional fue desarrollado teóricamente por

- ☒ Codd
- ☐ Elmasri
- ☐ Chen

**Obtenido = 1**

2.- (1 punto)

El concepto de relación matemática se adapta al modelo relacional

- ☐ permitiendo que las tuplas puedan repetirse y que las relaciones no estén obligadas a tener clave primaria
- ☐ creando los conceptos de clave alternativa y clave ajena

- ☒ asignando un nombre simbólico a los componentes de las tuplas de la relación

Obtenido = 1

3.- (1 punto)

La definición de relaciones (tablas) en una BD relacional establece las propiedades (del sistema de información que representan)

- ☐ de registros
- ☒ estáticas
- ☐ dinámicas

Obtenido = 1

4.- (1 punto)

La definición de tablas en un SGBD relacional constituye

- ☐ el estado de la base de datos
- ☒ el esquema de la base de datos
- ☐ la cardinalidad de la base de datos

Obtenido = 1

5.- (1 punto)

Si como producto de la adaptación del concepto de relación matemática al modelo relacional decimos que la relación tiene intensión y extensión, la segunda se define como

- ☐ un conjunto de dominios no necesariamente disjuntos
- ☒ el conjunto de n-tuplas, donde cada tupla es un conjunto de pares (nombreAtributo: valor)
- ☐ un conjunto de nombres de atributos distintos, cada uno de ellos asociado a su dominio correspondiente

Obtenido = 1

6.- (1 punto)

De dominios en el modelo relacional y tipos de datos en lenguajes de programación, sistemas de gestión de bases de datos, etc.

- ☐ son lo mismo
- ☐ los dominios se usan en el cálculo relacional de dominios y los tipos de datos en el de tuplas
- ☒ los tipos de datos son casos particulares de dominios

Obtenido = 1

7.- (1 punto)

Para adaptar el concepto de relación matemática al modelo relacional, Codd tuvo que

- ☒ poner nombre a los dominios que constituyen la relación
- ☐ poner orden a los dominios que forman la relación
- ☐ proponer el álgebra relacional como un lenguaje de manipulación de datos

Obtenido = 1

8.- (1 punto)

Una tabla en el modelo relacional

- ☐ cada columna debe contener un identificador
- ☒ no puede tener filas duplicadas
- ☐ define claramente el orden de sus filas

Obtenido = 1

9.- (1 punto)

Un dominio, en la teoría del modelo relacional es

- ☒ un conjunto de valores escalares
- ☐ un producto cartesiano de n valores
- ☐ nada más que un tipo de datos de los habituales en los lenguajes de programación

Obtenido = 1

10.- (1 punto)

la definición de la relación matemática por intensión, después de su adaptación al modelo relacional es equivalente a su

- ☐ contenido
- ☐ cardinalidad
- ☒ esquema

Obtenido = 1

11.- (1 punto)

Los dominios que definen la relación como estructura del modelo relacional no son tipos de datos estándar de computación,

- ☐ los dominios son tablas que se definen como un subconjunto del producto cartesiano de n tipos de datos.
- ☐ los dominios son un subconjunto de los tipos de datos.
- ☒ si acaso se pueden reformular como tipos de datos definidos por el usuario.

Obtenido = 1

12.- (1 punto)

El concepto matemático de tupla, como consecuencia del concepto de relación matemática (antes de su adaptación al modelo relacional), implica que

- ☐ no existe un orden entre las componentes de la tupla.
- ☐ se puede referenciar una componente por su posición dentro de la tupla o por su nombre.
- ☒ sólo existe una forma de referenciar una componente dentro de la tupla.

Obtenido = 1

13.- (1 punto)

El modelo relacional no recoge el concepto de

- ☐ clave primaria.

☒ atributo multivaluado.

☐ agregación.

Obtenido = 1

14.- (1 punto)

El trabajar con un SGBD que siga el Modelo Relacional fielmente nos garantiza

☐ que no hay redundancia de información.

☒ que en las tablas no hay tuplas duplicadas.

☐ la posibilidad de utilizar columnas multivaluadas.

Obtenido = 1

15.- (1 punto)

Para el Modelo Relacional, la no duplicidad de tuplas

☐ garantiza siempre la existencia de una clave primaria y una clave alternativa distintas.

☒ es una restricción implícita por el tipo de estructura en la que se basa el modelo.

☐ es una restricción que se deriva del hecho de que en toda relación deba existir siempre al menos una clave candidata.

Obtenido = 1

16.- (1 punto)

La relación matemática, antes de su adaptación al modelo relacional



- ☐ tiene ordenadas sus tuplas y admite duplicados porque finalmente ha de traducirse a una tabla
- ☐ no tiene orden entre las componente de sus tuplas ni duplicados
- ☒ no tiene orden entre sus tuplas ni duplicados

Obtenido = 1

17.- (1 punto)

La intensión de una relación se refiere

- ☐ a los datos que puede albergar en un instante de tiempo determinado
- ☒ a su esquema
- ☐ a su cardinalidad

Obtenido = 1

Enviar

[Volver](#)

# Examen T3 MR A2

**Calificación del examen = 20 (de 20)**

1.- (1 punto)

Las correspondencias entre clases  $\text{Card}(A, r) = (2,3)$

- ☒ obligan a que cada miembro de A se vincule con dos o tres miembros de otra clase de objetos
- ☐ indican una generalización solapada en dos de sus tres especializaciones
- ☐ no se pueden dar en ningún modelo de datos

**Obtenido = 1**

2.- (1 punto)

Una especificación de correspondencia entre clases  $\text{Card}(T,x) = (1,N)$

- ☐ indica una generalización parcial y solapada
- ☒ es imposible en un esquema de bases de datos relacionales

☐ se da cuando hay 2 claves ajenas, cada una en una tabla distinta, que "trabajan" para la misma relación "x"

Obtenido = 1

3.- (1 punto)

La integridad referencial en un SGBD relacional

☒ se cumple si toda la clave ajena es nula o ningún atributo de la clave es nulo y la referencia es válida.

☐ es la restricción que garantiza la no duplicidad de tuplas.

☐ se cumple en general para las restricciones de valor de los atributos.

Obtenido = 1

4.- (1 punto)

Si todo valor de clave ajena ha de aparecer en la tabla a la que hace referencia, nos estamos refiriendo a

☒ integridad referencial.

☐ restricciones de cardinalidad mínima uno.

☐ integridad de clave

Obtenido = 1

5.- (1 punto)

Una clave ajena en el modelo relacional

☐ es un tipo de clave candidata.

☒ indica una asociación entre objetos.

☐ nunca podrá estar formada por todos los atributos de una relación.

Obtenido = 1

6.- (1 punto)

Una restricción de correspondencia entre clases de objetos de cardinalidad mínima 3

☒ nos dice que la ocurrencia del objeto estará presente en la agregación al menos tres veces.

☐ indica la existencia de tres claves ajenas.

☐ no es real, nunca se puede dar en ningún sistema de información.

Obtenido = 1

7.- (1 punto)

Una generalización total y disjunta

☐ no se puede representar en ningún modelo de datos

☐ no existe, las generalizaciones sólo pueden ser parciales y solapadas

☒ no se puede representar en el modelo relacional

Obtenido = 1

8.- (1 punto)

Una clave candidata puede contener nulos

☐ si no es, además, clave alternativa o primaria

☐ si está compuesta por más de un atributo

☒ nunca

Obtenido = 1

9.- (1 punto)

Si R es una relación compuesta por tres atributos R(A,B,C), que las claves candidatas sean irreducibles significa que

☒ (A,C) y (A, B) pueden ser ambas claves candidatas

☐ (A,B) no puede ser clave candidata si (A) es clave ajena

☐ (A,B,C) no puede ser clave candidata

Obtenido = 1

10.- (1 punto)

En el modelo relacional, una clave primaria puede ser al mismo tiempo

☐ alternativa

☐ multivaluada

☒ clave ajena

Obtenido = 1

11.- (1 punto)

Una clave candidata (o parte de ella)

- ☒ puede contener cadenas vacías
- ☐ no puede contener ni valores nulos ni cadenas vacías
- ☐ puede contener valores nulos

Obtenido = 1

12.- (1 punto)

En una tabla en el modelo relacional

- ☐ sólo pueden existir dos claves candidatas (una primaria y otra alternativa)
- ☒ una clave candidata puede estar compuesta por varios atributos
- ☐ sólo puede existir una clave candidata

Obtenido = 1

13.- (1 punto)

La integridad de clave

- ☒ no permite nulos en ninguna de las columnas de la clave primaria
- ☐ se aplica a la clave primaria pero no a la clave alternativa
- ☐ exige que la clave ajena sea complemente nula o contenga un valor previamente almacenado en una clave primaria

Obtenido = 1

14.- (1 punto)

Si una tabla tiene 4 columnas, la cantidad máxima posible de claves candidatas es

☐ 4

☒ 6

☐ 8

Obtenido = 1

15.- (1 punto)

Al hablar de claves en el Modelo Relacional

☐ una clave candidata de una relación siempre debe ser también clave ajena.

☐ una clave ajena de una relación R debe coincidir en el número de columnas con el de la clave primaria de R.

☒ una clave ajena de una relación R puede ser también la clave primaria de R.

Obtenido = 1

16.- (1 punto)

Si una relación se define en función de sus atributos como R(a, b, c) y (a, b) es clave primaria,

☐ podría tener cuatro claves candidatas: la primaria antes mencionada y las alternativas (a, c), (b, c) y (c).

☐ ya no se pueden definir más claves candidatas.

☒ tendrá como máximo tres claves candidatas.

Obtenido = 1

17.- (1 punto)

Si una relación tiene más de una clave candidata

☐ tendrán que coincidir todas las claves candidatas en número de columnas.

☐ tendrá también más de una clave ajena.

☒ cualquiera de las claves candidatas sirve para identificar las tuplas de la misma.

Obtenido = 1

18.- (1 punto)

Sea  $R(a, b, c, d)$  una relación (tabla) con cuatro atributos (columnas) en la que se obvian otras definiciones necesarias.

☒ R puede tener como claves candidatas  $(a,c)$ ,  $(b,c)$  y  $(d)$

☐ R no puede tener más de 4 claves candidatas

☐ R puede tener como claves candidatas  $(a,c)$ ,  $(b,c)$  y  $(c,d,a)$

Obtenido = 1

19.- (1 punto)

Una clave candidata puede contener nulos

☐ si también es clave ajena



☒ nunca

☐ siempre que sea clave alternativa y no clave primaria

Obtenido = 1

20.- (1 punto)

Cuando decimos que no puede haber nulos en una clave primaria nos estamos refiriendo a

☒ integridad de clave

☐ integridad referencial

☐ integridad principal

Obtenido = 1

Enviar

[Volver](#)

# Examen T3 MR B

**Calificación del examen = 35 (de 35)**

1.- (1 punto)

Una relación 1:1

- ☐ una única tabla sin claves ajenas pero con una clave primaria y otra alternativa
- ☒ dos tablas más una tercera que aloja 2 claves ajenas a cada una de las anteriores, una como clave primaria y la otra como alternativa
- ☐ son dos tablas con una clave ajena en una de ellas que es, al mismo tiempo, clave alternativa

**Obtenido = 1**

2.- (1 punto)

Si tengo dos tablas definidas como

A(a,b) CP(a) CAj(b) -> B VNN(a)

B(c,d) CP(c) CAj(d) -> A VNN(d)

- ☐ B.d no puede ser clave ajena porque la clave primaria a la que apunta se llama "a", A.a.

☒ tenemos 2 relaciones 1:N con restricción de existencia independientes:  $\text{Card}(A,r)=(0,N)$ ,  $\text{Card}(B,r)=(1,1)$  y  $\text{Card}(A,s)=(1,1)$ ,  $\text{Card}(B,s)=(0,N)$ .

☐ se trata de una relación muchos a muchos con doble restricción de existencia,  $\text{Card}(A,r)=(1,N)$ ,  $\text{Card}(B,r)=(1,N)$ .

Obtenido = 1

3.- (1 punto)

Una relación 1:1 con una restricción de existencia

☐ una única tabla sin claves ajenas pero con una clave primaria y otra alternativa

☐ dos tablas más una tercera que aloja 2 claves ajenas a cada una de las anteriores, una como clave primaria y la otra como alternativa

☒ son dos tablas con una clave ajena en una de ellas que es, al mismo tiempo, clave alternativa

Obtenido = 1

4.- (1 punto)

Una relación 1:M con una restricción de existencia

☒ son dos tablas con una clave ajena en una de ellas que es, al mismo tiempo, valor no nulo

☐ son dos tablas con una clave ajena en una de ellas que es, al mismo tiempo, clave alternativa

☐ dos tablas más una tercera que aloja 2 claves ajenas a cada una de las anteriores, las dos componiendo la clave primaria

Obtenido = 1

5.- (1 punto)

Una tercera tabla con dos claves ajenas que componen la clave primaria

- ☐ es una relación uno a uno
- ☐ es una relación uno a muchos
- ☒ es una relación muchos a muchos

Obtenido = 1

6.- (1 punto)

Una tercera tabla con dos claves ajenas, una de ellas clave primaria y la otra alternativa

- ☐ es una relación muchos a muchos
- ☐ es una relación uno a muchos
- ☒ es una relación uno a uno

Obtenido = 1

7.- (1 punto)

Una clave ajena que hace referencia a otra tabla, sin restricciones adicionales

- ☐ es una relación muchos a muchos
- ☒ es una relación uno a muchos
- ☐ es una relación uno a uno

Obtenido = 1

8.- (1 punto)

Si EMPLEADO y DEPARTAMENTO se relacionan de forma que un empleado sólo puede trabajar en un departamento como máximo, en modelo relacional se representaría como

- ☐ una clave ajena en DEPARTAMENTO
- ☐ una tercera tabla con una clave ajena a EMPLEADO que sea, al mismo tiempo, clave alternativa.
- ☒ una clave ajena en EMPLEADO

Obtenido = 1

9.- (1 punto)

Si 2 tablas se relacionan mediante una clave ajena en una de ellas que es al mismo tiempo clave alternativa, estamos hablando de

- ☒ una relación uno a uno con restricción de existencia
- ☐ una relación uno a muchos con restricción de existencia
- ☐ una relación uno a uno

Obtenido = 1

10.- (1 punto)

Si MARINO capitanea uno y sólo un BARCO, mientras que los BARCOS pueden NO tener capitán pero como mucho UN capitán

- ☐ en la tabla MARINO hay una clave ajena sin restricciones

☒ en la tabla MARINO hay una clave ajena que es, al mismo tiempo, clave alternativa

☐ en la tabla CAPITANEA hay dos claves ajenas, a MARINO y BARCO respectivamente, que componen la clave primaria

Obtenido = 1

11.- (1 punto)

Si una tabla A tiene una relación con otra tabla y la clave ajena está definida en en esa otra tabla

☐ necesariamente tiene una restricción de existencia.

☒ nunca podrá tener restricción de existencia.

☐ tendrá restricción de existencia si esa clave ajena no admite nulos.

Obtenido = 1

12.- (1 punto)

Si no necesito que los coches tengan propietario pero sólo pueden tener uno, los propietarios sin restricciones

☐ pondré 2 claves ajenas en una tercera tabla.

☐ pondré la clave ajena en propietarios.

☒ pondré la clave ajena en coches.

Obtenido = 1

13.- (1 punto)

QUIRÓFANO(código, sala)

CP(código)

MÉDICO(dni, nombre, qfno)

CP(dni)

CAj(qfano) >> QUIRÓFANO

Card(MÉDICO, sustituye) =

Card(QUIRÓFANO, sustituye) =

Obtenido = 1

14.- (1 punto)

QUIRÓFANO(código, sala, doctor)

CP(código)

CAj(doctor) >> MÉDICO

VNN(doctor)

MÉDICO(dni, nombre)

CP(dni)

Card(QUIRÓFANO, manda) =

Card(MÉDICO, manda) =

Obtenido = 1

15.- (1 punto)

QUIRÓFANO(código, sala)

CP(código)

MÉDICO(dni, nombre)

CP(dni)

OPERA(qfno,doctor)

CP(qfano,doctor)

CAj(doctor) >> MÉDICO

CAj(qfano) >> QUIRÓFANO

Card(QUIRÓFANO, opera) =

Card(MÉDICO, opera) =

Obtenido = 1

16.- (1 punto)

QUIRÓFANO(código, sala)

CP(código)

MÉDICO(dni, nombre)

CP(dni)

SUPERVISA(qfno,doctor)



CP(qfano)  
CAIt(doctor)  
CAj(doctor) >> MÉDICO  
CAj(qfano) >> QUIRÓFANO

Card(QUIRÓFANO, supervisa) =

Card(MÉDICO, supervisa) =

Obtenido = 1

17.- (1 punto)  
QUIRÓFANO(código, sala)  
CP(código)  
CAIt(sala)  
CAj(sala) >> SALA

SALA(localizador, metros)  
CP(localizador)

Card(SALA, seubica) =

Card(QUIRÓFANO, seubica) =

Obtenido = 1

18.- (3 puntos)  
**PLANETA**(num,tamaño,coordenadas) CP(num)

**SATÉLITE**(snum,planeta) CP(planeta,snum)

CAj(planeta) >> PLANETA

**MISIÓN**(cod,objetivo,snum) CP(cod)

CAIt(objetivo,snum)

CAj(objetivo,sum) >> SATÉLITE

**VISITA**(planeta,misión,fecha,duración) CP(planeta,misión,fecha)

CAj(planeta) >> PLANETA

CAj(misión) >> MISIÓN

Card(PLANETA,tiene) = (0,N)

Card(SATÉLITE, tiene) = (1,1)

Card(MISIÓN, visita) = (0,N)

Card(PLANETA,visita) = (0,N)

Card(MISIÓN, va) = (1,1)

Card(SATÉLITE, va) = (0,1)

Obtenido = 3

19.- (3 puntos)

**PARCELA**(num,hm) CP(num)

**RÚSTICA**(num,gestiona) CP(num)

CAj(num) >> PARCELA

CAj(gestiona) >> VECINO VNN(gestiona)

**VECINO**(dni,nombre) CP(dni)  
**EQUIPO**(dni1,dni2) CP(dni1,dni2)  
CAj(dni1) >> VECINO  
CAj(dni2) >> VECINO

Card(RÚSTICA,es) = (1,1)

Card(PARCELA, es) = (0,1)

Card(VECINO, gestiona) = (0,N)

Card(RÚSTICA, gestiona) = (1,1)

Card(VECINO1, equipo) = (0,N)

Card(VECINO2, equipo) = (0,N)

Obtenido = 3

20.- (3 puntos)

**CIUDAD**(ciudad) CP(ciudad)  
**EQUIPO**(nombre,ciudad)  
CP(nombre)  
CAj(ciudad) >> CIUDAD  
VNN(ciudad)  
**CALENDARIO**(fecha) CP(fecha)  
**PARTIDO**(local, visitante, fecha)  
CP(local, visitante, fecha)

CAj(local) >> EQUIPO  
CAj(visitante) >> EQUIPO  
CAj(fecha) >> CALENDARIO

Card(EQUIPO, de) = (1,1)

Card(CIUDAD, de) = (0,N)

Card(EQUIPO1, partido) = (0,N)

Card(EQUIPO2, partido) = (0,N)

Card(PARTIDO, se\_juega\_el) = (1,1)

Card(CALENDARIO, se\_juega\_el) = (0,N)

Obtenido = 3

21.- (3 puntos)

**PLANETA**(num,tamaño,coordenadas) CP(num)

**SATÉLITE**(snum,planeta) CP(snum)

CAj(planeta) >> PLANETA

**MISIÓN**(cod,objetivo) CP(cod,objetivo)

CAj(objetivo) >> SATÉLITE

**VISITA**(planeta,misión,objetivo,fecha,duración)

CP(planeta)

CAlt(misión,objetivo)

CAj(planeta) >> PLANETA

CAj(misión,objetivo) >> MISIÓN

Card(PLANETA,tiene) = (0,N)

Card(SATÉLITE, tiene) = (0,1)

Card(MISIÓN, visita) = (0,1)

Card(PLANETA,visita) = (0,1)

Card(MISIÓN, va) = (1,1)

Card(SATÉLITE, va) = (0,N)

Obtenido = 3

22.- (3 puntos)

**RÚSTICA**(num) CP(num)

**PARCELA**(num,hm,cat) CP(num,cat)

CAj(cat) >> RÚSTICA

**VECINO**(dni,nombre) CP(dni)

**EQUIPO**(dni1,dni2,gestiona)

CP(dni1,dni2)

CAj(dni1) >> VECINO

CAj(dni2) >> VECINO

CAj(gestiona) >> VECINO VNN(gestiona)

Card(RÚSTICA,es) = (0,N)

Card(PARCELA, es) =

Card(VECINO, gestiona) =

Card(EQUIPO, gestiona) =

Card(VECINO1, equipo) =

Card(VECINO2, equipo) =

Obtenido = 3

23.- (3 puntos)

**EQUIPO**(nombre) CP(nombre)

**CIUDAD**(ciudad, equipo)

CP(ciudad)

CAlt(equipo)

CAj(equipo) >> EQUIPO

**CALENDARIO**(fecha) CP(fecha)

**PARTIDO**(local, visitante, fecha)

CP(local, visitante)

CAlt(fecha)

CAj(local) >> EQUIPO

CAj(visitante) >> EQUIPO

CAj(fecha) >> CALENDARIO

Card(EQUIPO, de) =

Card(CIUDAD, de) =

Card(EQUIPO1, partido) =

Card(EQUIPO2, partido) =

Card(PARTIDO, se\_juega\_el) =

Card(CALENDARIO, se\_juega\_el) =

Obtenido = 3

Enviar

[Volver](#)

# Examen T4 Lenguajes

**Calificación del examen = 18 (de 18)**

1.- (1 punto)

El álgebra relacional

- ☒ está basado en la teoría de conjuntos
- ☐ está basado en los lenguajes de primer orden
- ☐ está basado en el modelo relacional

**Obtenido = 1**

2.- (1 punto)

Para que dos relaciones sean compatibles en álgebra relacional

- ☒ deben tener el mismo grado
- ☐ deben tener la misma cardinalidad



☐ no es necesario que tengan ni la misma cardinalidad ni el mismo grado

Obtenido = 1

3.- (1 punto)

Las relaciones derivadas en álgebra relacional

☐ no existen las relaciones derivadas en el álgebra relacional

☐ no tienen nombre ni alias, ni tampoco nombre de columnas

☒ no tienen nombre ni alias

Obtenido = 1

4.- (1 punto)

Una relación derivada se define en el álgebra relacional como

☐ una relación muchos a muchos entre dos tablas.

☒ la relación, tabla, resultado de operar en AR.

☐ una relación con nombre.

Obtenido = 1

5.- (1 punto)

Al efectuar una proyección de álgebra relacional en una tabla por una columna que NO sea clave candidata

☐ puede provocar la aparición de duplicados de filas en la relación resultado.

☐ la proyección siempre se debe hacer con al menos una clave candidata en la lista de columnas especificada para la operación.

☒ dado que el resultado también es una relación, no es posible obtener duplicados de filas.

Obtenido = 1

6.- (1 punto)

Al efectuar un producto cartesiano de álgebra relacional de una tabla por si misma

☐ provoca de duplicados de tupla en la relación resultado, que no se darían si el producto fuera entre dos tablas diferentes.

☐ las columnas de la tabla derivada no tienen nombre.

☒ por definición, nunca produce duplicados de tuplas.

Obtenido = 1

7.- (1 punto)

El operador concatenación natural del álgebra relacional, para poderse ejecutar apropiadamente

☐ necesita atributos con el mismo dominio pero no necesariamente con el mismo nombre

☐ precisa de claves ajenas

☒ necesita atributos con el mismo dominio y con el mismo nombre

Obtenido = 1

8.- (1 punto)

En realidad, el operador división del álgebra relacional es una combinación de

- ☒ diferencias, producto cartesiano y proyecciones
- ☐ no es combinación de nada, es una primitiva
- ☐ selección, producto cartesiano y proyección

Obtenido = 1

9.- (1 punto)

La concatenación natural del álgebra relacional

- ☐ sólo se puede utilizar si entre las dos tablas operando existe una clave ajena de una a otra
- ☐ exige relaciones operando compatibles
- ☒ siempre se puede expresar como una secuencia de producto cartesiano, selección y proyección

Obtenido = 1

10.- (1 punto)

La selección en álgebra relacional

- ☐ da como resultado el conjunto de tuplas combinadas de dos relaciones
- ☐ obtiene un subconjunto de atributos de una relación
- ☒ obtiene un conjunto de tuplas en el que todas cumplen una condición establecida

Obtenido = 1

11.- (1 punto)

En álgebra relacional, el producto cartesiano entre dos relaciones necesita

- ☐ que éstas sean compatibles
- ☐ que haya algún atributo con el mismo nombre y dominio en las dos tablas
- ☒ no necesita nada, se puede hacer siempre que queramos

Obtenido = 1

12.- (1 punto)

El nombre de las columnas del resultado de realizar una unión en álgebra relacional

- ☐ son los mismos que los de la segunda relación.
- ☐ no importan porque las dos tablas deben tener los mismo nombres de columna y los mismos dominios en las mismas posiciones.
- ☒ son los mismos que los de la primera relación.

Obtenido = 1

13.- (1 punto)

En álgebra relacional, suponiendo que las columnas con el mismo nombre tienen los mismos dominios en cualquier tabla, y siendo  $T(a,b,c)$  y  $S(d,b,c,e)$ , si efectúo  $T[a,b,c] \text{ DIVIDIDO } (S[b,c])$ , el resultado es

- ☒ una relación con una columna,  $R[a]$ .
- ☐ una relación con 3 columnas  $R[a,b,c]$ .

☐ nada, esta expresión no es correcta.

Obtenido = 1

14.- (1 punto)

Una consulta en álgebra relacional que sea parecida a "dame TODOS los vendedores de la provincia de Alicante"

☐ necesita el operador DIVISIÓN.

☐ no se puede hacer porque precisa hacer la cuenta de filas que hay en la tabla, y el álgebra relacional está muy limitado para contar, ordenar, etc.

☒ no necesita el operador DIVISIÓN

Obtenido = 1

15.- (1 punto)

Uno de los indicadores de la importancia del álgebra dentro del modelo relacional es

☐ su potencia de cálculo.

☒ su uso en ciertos optimizadores de consultas.

☐ que SQL, por ejemplo, no ofrece operadores como la unión, intersección o diferencia.

Obtenido = 1

16.- (1 punto)

Una de las diferencias entre el álgebra relacional y SQL o los cálculos relacionales es que

☐ hay ciertas consultas a la base de datos que se pueden hacer en AR pero no en los otros.

☐ está mejor adaptado al modelo relacional que los otros.

☒ el AR es procedimental, establece una secuencia de operaciones, mientras que los otros son declarativos.

Obtenido = 1

17.- (1 punto)

El cálculo relacional es

☐ lógico

☒ declarativo

☐ procedimental

Obtenido = 1

18.- (1 punto)

El álgebra relacional es

☒ procedimental

☐ lógico

☐ declarativo

Obtenido = 1

Enviar



# Examen T5 Normaliza

**Calificación del examen = 17 (de 17)**

1.- (1 punto)

Para evitar redundancias en una base de datos relacional

- ☐ la política adecuada para mantener la integridad referencial es la de anular, aunque a veces es decisión del diseñador emplear otra política.
- ☒ se aplica el proceso de normalización hasta obtener relaciones en 3FN (en la mayoría de los casos).
- ☐ las claves ajenas deben tener prevista la política adecuada ante operaciones de borrado.

**Obtenido = 1**

2.- (1 punto)

La forma normal de Boyce-Codd en una relación R

- ☐ se cumple siempre que haya una única clave candidata
- ☒ se cumple cuando R está en tercera forma normal y tiene varias claves candidatas no solapadas



☐ no hace falta tratarla si R está ya en 3FN aunque no necesariamente en 2FN

Obtenido = 1

3.- (1 punto)

Un defecto de normalización en una base de datos relacional puede provocar anomalías

- ☐ al borrar la información de una tupla, ya que se pueden estar borrando tuplas de otras tablas involuntariamente
- ☐ al insertar información en una tabla, porque los datos insertados no se corresponderán con la realidad
- ☒ al modificar la información de una tabla, ya que un cambio simple de un dato podría afectar a varias tuplas.

Obtenido = 1

4.- (1 punto)

La dependencia funcional es

- ☐ otra forma de denominar a la integridad referencial
- ☒ una relación entre atributos de una tabla de tal forma que un valor de uno determina unívocamente el valor del otro
- ☐ una restricción del modelo que dice que las claves candidatas no admiten nulos, ni total ni parcialmente

Obtenido = 1

5.- (1 punto)

Una relación en tercera forma normal con 3 atributos puede tener, como máximo

- ☐ una clave candidata

☐ seis claves candidatas

☒ tres claves candidatas

Obtenido = 1

6.- (1 punto)

La normalización

☐ no es necesaria aunque sí recomendable

☒ se utiliza actualmente más como un criterio de calidad en el diseño

☐ es el mejor método para diseñar las bases de datos

Obtenido = 1

7.- (1 punto)

La primera forma normal garantiza que

☐ no existan dependencias funcionales directas

☐ no existan dependencias funcionales transitivas

☒ los dominios contienen valores atómicos

Obtenido = 1

8.- (1 punto)

Una tabla en el modelo relacional

- ☐ puede estar en tercera forma normal aunque no esté en primera
- ☐ puede estar en tercera forma normal aunque no esté en segunda
- ☒ sólo puede estar en tercera forma normal si lo está en primera y en segunda

Obtenido = 1

9.- (1 punto)

La forma normal de Boice-Codd

- ☐ es condición necesaria para que una tabla esté en tercera forma normal cuando hay más de una clave candidata
- ☒ se debe comprobar en tablas con dos o más claves candidatas
- ☐ se debe comprobar en tablas con una única clave candidata

Obtenido = 1

10.- (1 punto)

En la normalización de tablas en modelo relacional se habla de atributos *primos* y *no primos*

- ☒ solo los atributos no primos pueden causar problemas de 2FN y 2FN.
- ☒ los atributos no primos son los que no pertenecen a ninguna clave candidata.
- ☐ puede haber tablas sin ningún atributo primo.

Obtenido = 1

11.- (1 punto)

La segunda forma normal persigue que

- ☐ no haya claves alternativas
- ☒ no haya dependencias funcionales incompletas
- ☐ no haya dependencias funcionales transitivas

Obtenido = 1

12.- (1 punto)

La tercera forma normal busca que

- ☒ no haya dependencias funcionales transitivas
- ☐ no haya claves alternativas
- ☐ no haya dependencias funcionales incompletas

Obtenido = 1

13.- (1 punto)

La forma normal Boyce-Codd intenta eliminar

- ☒ aquellos determinantes que no son clave candidata
- ☐ las dependencias funcionales cíclicas (de A a b y de B a A, simultáneamente)
- ☐ las claves alternativas

Obtenido = 1

14.- (1 punto)

Cuando decimos que un atributo es "primo" es porque

- ☐ está "aislado", no depende ni determina funcionalmente de o a ningún otro atributo
- ☐ también es un determinante
- ☒ pertenece a una clave candidata

Obtenido = 1

15.- (1 punto)

Si A determina funcionalmente a B, B a C, y C a A, siendo estos los únicos atributos de la relación a normalizar, ¿cuántas claves candidatas tiene la tabla?

- ☐ Dos
- ☐ Una
- ☒ Tres

Obtenido = 1

16.- (1 punto)

Si en una tabla no hay dependencias funcionales entre sus atributos

- ☐ habrá tantas claves candidatas como columnas tenga la tabla
- ☒ esa tabla solo tiene una clave candidata

☐ eso nunca ocurrirá en modelo relacional: toda tabla tiene al menos una dependencia funcional

Obtenido = 1

17.- (1 punto)

Durante el proceso de normalización de una tabla

☐ a veces se generan dos claves primarias en la misma tabla

☒ es imposible que se generen, en la misma tabla, dos claves ajenas solapadas en atributos

☐ puede que se generen, en la misma tabla, dos claves ajenas solapadas en atributos (CAj(A)->tabla1; CAj(A,B)->tabla2)

Obtenido = 1

Enviar

[Volver](#)