

Defina os seguintes termos: Bases de Dados, Sistema de Gestão de Bases de Dados identificando os seus componentes, Metadados.
<p>BASE DE DADOS: coleção partilhada de dados logicamente relacionados e a descrição desses dados, desenhado para satisfazer a necessidade de informação de uma organização.</p> <p>SISTEMA DE GESTÃO DE BASE DE DADOS (SGBD): sistema de software que permite aos utilizadores definir, criar, manter e controlar o acesso à base de dados. Componentes: hardware, software, dados, processamento e pessoas.</p> <p>METADADOS: repositório de informação que descreve os dados na base de dados. Este disponibiliza a descrição dos dados para obtermos aplicações independentes.</p>
Descreve os 5 componentes do ambiente de um SGBD e mostre como elas se relacionam.
<p>Primeiramente, existe o HARDWARE que é a parte física deste sistema, compreendendo toda a estrutura “real” onde assenta o SGBD.</p> <p>De seguida temos o SOFTWARE que consiste em toda a parte lógica do sistema, como o sistema operativo, que permite ao SGBD funcionar corretamente.</p> <p>No centro temos os DADOS que fazem a ligação entre as duas extremidades.</p> <p>No que diz respeito à segunda extremidade do sistema este contém os PROCEDIMENTOS que permitem interagir com a BD e temos também as PESSOAS, o “componente” final que interage com a BD.</p>
Descreva o conceito de independência de dados e a sua importância num ambiente de bases de dados.
O conceito de INDEPENDÊNCIA DE DADOS significa que mudanças relacionadas aos níveis inferiores (lógico ou físico), não afetam os níveis superiores.
Descreva o papel de um gestor de BD numa solução baseada na mesma.
<p>Numa solução que faça uso de uma BD é importante que exista um SISTEMA DE GESTÃO da mesma uma vez que sem este a criação da BD é impossível.</p> <p>Um SGBD é responsável por controlar todas as operações de uma BD se este não existir a BD também não existe.</p> <p>Qualquer administrador de BD deveria ter um conhecimento profundo do sistema de gestão uma vez que tudo o que executar sobre a BD terá de ser feito com recurso ao SGBD.</p>
Descreva em que situações será preferível a abordagem de Sistemas de Ficheiros comparativamente à abordagem de Sistemas de Bases de Dados, tendo em atenção as principais desvantagens da utilização de um SGBD.
Quando a quantidade de informação armazenada é baixa e tem o propósito de servir apenas um departamento é preferível um sistema menos complexo tal como um SISTEMA DE FICHEIROS . Para além do tamanho e complexidade serem baixos o custo comparativamente a um SGBD é também muito inferior e em caso de falha o impacto é inferior.
Descreva as principais características de um Sistema BD e faça a comparação com os Sistemas Baseados em Ficheiros.
<p>Os SISTEMAS DE BD surgiram com o intuito de colmatar as limitações dos SISTEMAS BASEADOS EM FICHEIROS.</p> <p>Nos SISTEMAS DE BD existe uma centralização da informação que pode ser acedida em simultâneo por diversos utilizadores, eliminando assim a separação e isolamento dos dados, bem como a redundância de informação, características dos SISTEMAS BASEADOS EM FICHEIROS.</p> <p>Para além disso, num SISTEMA DE BD existe a independência entre informação e as aplicações que fazem uso dela, mais uma vez melhorando o SISTEMA DE FICHEIROS, onde as aplicações eram desenhadas para um tipo específico de ficheiros.</p>

Descreva quais as principais características na abordagem de Sistemas de Bases Dados e compare com a abordagem de Sistemas de Ficheiros.

Enuncie e explique sucintamente as principais vantagens e desvantagens de um SGBD.

Os Sistemas de BD surgiram com o intuito de colmatar as limitações dos Sistemas de Ficheiros. Nos Sistemas de BD existe uma centralização da informação, que pode ser acedida em simultâneo por diversos utilizadores, eliminando assim a separação e isolamento dos dados, bem como a redundância de informação que é uma característica dos Sistemas de Ficheiros.

Para além disso, num Sistema de BD existe independência entre a informação e as aplicações que fazem uso dela, mais uma vez melhorando o Sistema de Ficheiros, onde as aplicações são desenhadas para um tipo específico de ficheiros.

Vantagens:

- *Controlo sobre a redundância de dados*
- *Consistência de dados*
- *Partilha de dados*
- *Mais segurança*
- *Mais produtividade*
- *Mais concorrência*
- *Mais informação tendo em conta os mesmos dados*
- *Melhoria na manutenção através da independência de dados*
- *Integridade de dados melhorada*
- *Implica uso de standards*
- *Acessibilidade aos dados e rapidez de resposta melhorados*
- *Requisitos conflituosos balanceados*
- *Serviços de cópias de segurança e de recuperação de falhas melhoradas*
- *Economia de escala*

Desvantagens:

- *Complexidade*
- *Tamanho*
- *Custo do SGBD*
- *Custo do hardware acrescido*
- *Custo de conversão*
- *Performance*
- *Maior impacto em caso de falha*

Identifique as funções que um SGBD deve satisfazer.

- *Armazenamento, Pesquisa e Atualização de Dados.*
- *Dicionário de Dados.*
- *Suporte a Transações.*
- *Serviços de Controlo de Concorrência.*
- *Serviços de Recuperação.*
- *Serviços de Autenticação.*
- *Suporte a Comunicação de Dados.*
- *Serviços de Integridade.*
- *Serviços que promovam a Independência de Dados.*
- *Utilitários.*

Descreva as 3 gerações de SGBD.

Os SGBD podem ser divididos em 3 gerações:

Inicialmente, existia o **MODELO HIERÁRQUICO** que apesar de trazer algumas melhorias sobre o Sistema de Ficheiros, apresenta falhas na independência dos dados e precisava de programas complexos, para processar a informação.

Posteriormente surgiu o **MODELO RELACIONAL** que veio resolver muitos problemas da primeira geração.

A 3ª geração compreende **SGBD ORIENTADOS A OBJECTOS** e **OBJECTOS-RELACIONAL**, empregando assim um novo tipo de paradigma à gestão de BD, simplificando e resolvendo mais algumas questões deixados pelas gerações anteriores.

O que são sublinguagens de dados? Porque são importantes?

São a forma de comunicação existente com a BD. Geralmente dividem em duas vertentes.

Existe a linguagem de **DEFINIÇÃO DE DADOS (DDL)** que permite a implementação da própria BD, bem como de todas as relações existentes e possíveis restrições de integridade associados.

Por outro lado, existe a linguagem de **MANIPULAÇÃO DE DADOS (DML)** usado após a criação da BD que permite inserir, eliminar, atualizar e consultar a informação presente na BD.

Estas linguagens são importantes porque é através delas que é possível dizer ao SGBD o que este deve fazer na BD.

Apresente as diferenças entre DDL e DML.

São sublinguagens aplicadas à BD que se completa.

DDL permite a criação da própria BD, das relações que a constituem, das regras de integridade sobre os dados, etc.

DML é usada após a criação da BD para a inserção, atualização, remoção e consulta de informação.

Apresente as diferenças entre DML Procedimentais e DML Não Procedimentais.

A grande diferença entre as DML referidas é o seu propósito final, uma vez que as **DML PROCEDIMENTAIS** são utilizadas quando é necessário trabalhar a forma como os dados são apresentados e as **DML NÃO PROCEDIMENTAIS** apenas se “preocupam” com quais os dados que serão apresentados.

Discuta a função de um System Catalog.

É fundamental num SGBD pois contém metadado, ou seja, informação sobre a informação, que deve estar acessível aos utilizadores uma vez que permite a esses mesmos utilizadores uma melhor compreensão sobre a estrutura da BD, desde os esquemas às aplicações, passando pelos nomes das relações às regras de integridade de informação.

Descreva o propósito da criação de índices sobre relações e em que situações será preferível criá-los ou não.

Os índices servem para obter informação mais rapidamente, pois sem índice o SGBD lê toda a tabela para localizar a informação pretendida. Já com índice, o SGBD percorre o índice para descobrir onde obter os dados e em seguida aceder diretamente aos locais para obter os mesmos.

O que é controlo de concorrência e porque é que é importante num SGBD.

É algo que garante que quando existem vários utilizadores a alterar a informação na BD essas alterações são realizadas corretamente.

Um mecanismo deste género é fundamental num SGBD pois só assim é possível o acesso partilhado à informação.

Explique o conceito de Database Schema e descreva os três tipos de schema numa BD.
<p>Um DATABASE SCHEMA consiste numa descrição da estrutura de BD sendo possível encontrar três tipos diferentes cada um correspondente a um nível da arquitetura ANSI/SPARC.</p> <p>O SCHEMA INTERNO concentra todas as informações relativas ao armazenamento físico da BD.</p> <p>O SCHEMA CONCEPTUAL no qual se encontra todo o desenho “virtual” da BD, todas as relações e entidades que existem e o seu modo de relacionamento.</p> <p>Por fim existe o SCHEMA EXTERNO que diz respeito ao modo como a informação é apresentada ao utilizador final, podendo haver vários schemas externos para diferente utilizadores.</p>
Discuta a função e importância do modelo conceptual.
<p>A MODELAÇÃO CONCEPTUAL de uma BD é a fase inicial da sua implementação e a mais importante.</p> <p>É nesta modelação que são definidos as relações, entidades, relacionamentos e regras de integridade, ignorando detalhes de implementação e todas as definições deverão ser feitas com cuidado uma vez que se o modelo de BD estiver corretamente desenhado, a sua implementação não funciona em qualquer SGBD.</p> <p>Fazendo um paralelo à programação o modelo conceptual pode ser o algoritmo de uma BD.</p>
Identifique quais os objetivos da arquitetura de três níveis ANSI/SPARC.
<p>A ARQUITETURA DE TRÊS NÍVEIS ANSI/SPARC tem como objetivo que todos os utilizadores possam aceder aos mesmos dados, sendo que estes não necessitam de saber pormenores do armazenamento físico da base de dados, outro objetivo desta arquitetura é que cada utilizador tenha uma vista que é imune a alterações feitas noutras vistas.</p> <p>Em relação ao administrador da BD, esta arquitetura permite que este possa alterar a estrutura de armazenamento da BD sem que esta afete as vistas dos utilizadores, podendo também alterar a estrutura conceptual da BD sem afetar os utilizadores.</p> <p>Outro objetivo é que a estrutura interna da base de dados não seja afetada pelas alterações efetuadas no armazenamento físico.</p>
Compare a arquitetura cliente-servidor de dois níveis com a de três níveis e identifique justificando qual a mais adequada para a Web.
<p>Numa ARQUITETURA CLIENTE-SERVIDOR de duas camadas temos apenas um servidor responsável pelo SGBD e um cliente que corre as aplicações.</p> <p>Numa ARQUITETURA COM TRÊS CAMADAS, passa a existir um servidor de base de dados, servidor de aplicação e uma interface de utilizador que corre no cliente, fazendo desta arquitetura mais aplicável ao modelo web, uma vez que o servidor de aplicação poderá estar incluído num servidor web e mesmo a interface do utilizador poderá ser baseado na web, onde os clientes podem aceder a partir de um browser.</p> <p>Esta arquitetura dá assim suporte a acessibilidade das BD, uma vez que é possível aumentar o numero de servidores web, fazendo um balanceamento de carga.</p>
A arquitetura ANSI/SPARC identifica três níveis nos SGBD. Descreva pormenorizadamente o nível intermédio, identificando o seu nome, e o que se pretende que este nível represente.
<p>NÍVEL EXTERNO: visão de um utilizador sobre uma base de dados, descreve parte da base de dados que é relevante a um determinado utilizador;</p> <p>NÍVEL CONCEPTUAL: representa a união das vistas das bases de dados, descreve quais os dados armazenados na base de dados e quais as relações entre estes, no nível conceptual representam-se as entidades, os atributos, relacionamentos, restrições, informação semântica sobre os dados, informação sobre segurança e integridade;</p> <p>NÍVEL INTERNO: representação física da base de dados no computador, descreve como os dados são armazenados na base de dados;</p>

Identifique quais as propriedades das Relações no Modelo Relacional.
<p>Uma RELAÇÃO tem de ter um nome único no schema que identifica cada calculo da relação contem apenas um valor.</p> <p>Cada ATRIBUTO tem um nome distinto e os valores de cada um pertencem ao mesmo dominio. A ordem dos atributos não tem qualquer significado.</p> <p>Cada TUPLO é diferente e não existem duplicados. A sua ordem em termos teóricos não tem interesse, na pratica pode afetar a performance no acesso.</p>
Defina as duas principais regras de integridade no Modelo Relacional.
<p>A PRIMEIRA REGRA consiste na integridade da entidade, que garante que nenhum tuplo da relação possa ter o valor null na chave primária.</p> <p>A SEGUNDA REGRA consiste na integridade referencial que afirma que se um atributo for chave estrangeira numa relação só poderá assumir valor null as que estejam na tabela onde é chave primária.</p> <p>Pode ainda ser definida uma terceira regra, a regra da integridade geral, onde se implementam as várias restrições relativas ao negocio da BD.</p>
Na apresentação do modelo relacional foram enunciados alguns conceitos. Explique o que significa cada um dos seguintes termos: Relação, Atributo, Domínio, Tuplo, Grau, Cardinalidade.
<p>RELAÇÃO é uma tabela com colunas e linhas, apenas se aplica à estrutura lógica da base de dados, não à física.</p> <p>ATRIBUTO é o nome de uma coluna de uma relação.</p> <p>DOMÍNIO é um conjunto de valores permitidos para um ou mais atributos.</p> <p>TUPLO é uma linha de uma relação.</p> <p>GRAU é o número de atributos de uma relação.</p> <p>CARDINALIDADE é o número de tuplos de uma relação.</p>
Defina as cinco operações principais de álgebra relacional. Defina a Junção, Intersecção, Divisão através da utilização das cinco operações básicas.
<p>As cinco operações básicas da álgebra relacional são:</p> <ul style="list-style-type: none"> • SELEÇÃO: seleciona tuplos que satisfaçam à condição de seleção; • PROJEÇÃO: projeta as colunas solicitadas; • PRODUTO CARTESIANO: combina tuplos de duas relações; • UNIÃO: une duas tabelas. • DIFERENÇA: dá uma tabela com todas as linhas de A que não estão em B. <p>A JUNÇÃO de duas tabelas, A e B, dá uma tabela contendo todas as linhas de A ou B ou ambas.</p> <p>A INTERSECÇÃO dá uma tabela com as linhas comuns às tabelas A e B.</p> <p>A DIVISÃO mostra todos os valores de um atributo da tabela A que fazem referência a todos os valores de um atributo da relação B.</p>
Enuncie a definição de cada um dos seguintes termos: Chave Candidata, Chave Primária, Chave Estrangeira.
<p>CHAVE CANDIDATA é um ou mais atributos que identificam unicamente um tuplo na relação.</p> <p>CHAVE PRIMÁRIA consiste na chave candidata escolhida para a identificação dos tuplos.</p> <p>CHAVE ESTRANGEIRA é um atributo de uma relação que é chave candidata/primária numa outra relação e que permite relacionar tuplos de relações diferentes.</p>
Descreva os tipos de anomalias de atualização (dê exemplos) que podem ocorrer numa relação que contém dados redundantes.
<p>INSERÇÃO: inserir um funcionário num escritório que não existe.</p> <p>REMOÇÃO: ao apagar funcionário único, apaga também esse escritório e tudo o que está associado é perdido.</p> <p>MODIFICAÇÃO: ter informação duplicada e apenas atualizar parte desta informação, não atualizando a informação duplicada, cria inconsistência de dados.</p>

Explique o que significam cada uma das cláusulas utilizadas no comando SELECT.
<p>SELECT: obrigatório, atributos que aparecerão;</p> <p>FROM: obrigatório, tabela de origem;</p> <p>WHERE: condição, não podem ser usadas em funções de agregação;</p> <p>GROUP BY: agrupar por atributo;</p> <p>HAVING: filtrar grupos, podem ser funções de agregação;</p> <p>ORDER BY: ordenação, última cláusula, por defeito ascendente.</p>
Explique a importância e aplicação da cláusula WHERE em UPDATE e DELETE.
<p>A CLAUSULA WHERE é importante e vantajosa em UPDATE e DELETE porque permite selecionar o(s) registo(s) que deverão sofrer a atualização ou remoção, uma vez que estes comandos na ausência da CLAUSULA WHERE aplicam-se a todos os registos da tabela.</p>
Explique como funciona a cláusula GROUP BY. Qual a diferença entre a cláusula WHERE e HAVING?
<p>A cláusula GROUP BY é usada para obter sub-totais.</p> <p>Quando o SELECT e GROUP BY usados em conjunto: cada coluna na lista da expressão SELECT tem que ter um só valor por grupo, e a cláusula SELECT apenas pode conter:</p> <ul style="list-style-type: none"> • nomes de colunas; • funções de agregação; • constantes; • expressões contendo combinações das anteriores. <p>Todos os nomes das colunas existentes na expressão SELECT tem que estar também na cláusula GROUP BY com exceção dos nomes que são apenas usados nas funções de agregação.</p> <p>Se WHERE é usado em conjunção com GROUP BY, WHERE é aplicado primeiro, depois os grupos são formados das linhas resultantes que satisfaçam o predicado.</p> <p>A cláusula HAVING usa-se em conjunto com a cláusula GROUP BY para restringir os grupos que aparecem na tabela final. Os nomes das colunas que aparecem na cláusula HAVING têm de aparecer também na cláusula GROUP BY ou estarem contidos numa função de agregação.</p> <p>A DIFERENÇA entre o WHERE e o HAVING é que este filtra linhas individuais enquanto que HAVING filtra grupos.</p>
Quais as restrições aplicadas ao uso de funções de agregação no comando SELECT? De que forma os valores nulos (NULL) afetam as funções de agregação?
<p>Existem 5 funções de agregação no comando SELECT, que operam sobre uma única coluna e devolvem um único valor.</p> <p>Das 5 funções, SUM() e AVG() apenas operam em valores numéricos.</p> <p>Na cláusula SELECT se existir uma função de agregação e a cláusula GROUP BY não é utilizada, não pode existir no SELECT mais nenhuma referência a uma coluna da tabela, exceto aquela cuja a função de agregação usa.</p> <p>Em relação aos VALORES NULL, tirando a função COUNT() todas as outras operam em valores não nulos. Por fim para eliminar duplicados pode-se usar o DISTINCT.</p>
Explique o que entende por Integridade Referencial.
Identifique e descreva quais as ações que se podem utilizar nas subcláusulas ON DELETE e ON UPDATE.
<p>A INTEGRIDADE REFERENCIAL preserva as relações definidas entre tabelas quando linhas são criadas ou excluídas.</p> <p>As subcláusulas que podem ser usadas ON DELETE e em ON UPDATE são:</p> <ul style="list-style-type: none"> • CASCADE: apaga a linha da tabela pai e linhas correspondentes das tabelas filhas, e assim sucessivamente em cascata. • SET NULL: apaga a linha da tabela pai e muda todas as colunas FK na tabela filha para NULL. Só é válido se as colunas FK não estiverem a NOT NULL. • SET DEFAULT: apaga a linha da tabela pai e muda cada componente da FK da tabela filha para o valor default especificado. Só é válido se houver um valor DEFAULT especificado para as colunas FK. • NO ACTION: rejeita a operação da tabela pai. Default.

Como poderão ser combinadas os resultados de duas queries?
Poderão ser combinadas através das operações de UNION, INTERSECT, EXCEPT com a única condicionante de serem ambas as relações Union Compatible , ou seja, apresentam o mesmo número de atributos e que os atributos correspondentes nas duas tabelas têm o mesmo domínio.
Qual a diferença entre uma subquery e uma junção? Em que situações não é possível usar uma subquery?
Uma SUBQUERY é uma query ou uma subseleção embebida noutra query, enquanto a JUNÇÃO é a união entre várias tabelas. Não é possível utilizar uma SUBQUERY como um operador numa expressão.
Diferencia entre os 3 tipos de subqueries.
Dentro das subqueries é possível distinguir 3 tipos diferentes: Uma SUBQUERY ESCALAR que devolve um valor singular. Uma SUBQUERY DE LINHA que devolve apenas um tuplo. Uma SUBQUERY DE TABELA que retorna uma relação.

Descreva as diferenças entre as cinco operações de junção: Theta Join, Equijoin, Natural Join, Outer Join e Semijoin. Dê exemplos para suportar a sua resposta.
THETA JOIN é usado quando os atributos de junção não são homónimos. EQUIJOIN é usado quando a condição contém apenas igualdade. NATURAL JOIN é a junção normal feita através de um produto cartesiano seguido de uma seleção. OUTER JOIN serão mostrados toda a informação de uma relação e eventualmente do outro caso haja junção. SEMIJOIN apenas aparecerão os tuplos de uma relação (que participam na junção).

THETA JOIN EXEMPLO				
TABELA ESTUDANTE			TABELA CADEIRA	
ID	NOME	TURMA	TURMA	CADEIRA
1	Carlos	111	111	Base de Dados
			111	LP2
2	Teixeira	222	222	Física
			222	Engenharia
Álgebra Relacional: Detalhes_Estudante <- ESTUDANTE ⋈ (Estudante.Turma = Cadeira.Turma) CADEIRA				
TABELA DETALHES DO ESTUDANTE				
ID	NOME	TURMA	TURMA	CADEIRA
1	Carlos	111	111	Base de Dados
1	Carlos	111	111	LP2
2	Teixeira	222	222	Física
2	Teixeira	222	222	Engenharia

EQUIJOIN EXEMPLO						
TABELA ESTUDANTE				TABELA PROFESSOR		
ID	NOME	TURMA	IDADE	ID_Prof	NOME	CLASSE
1	Carlos	111	23	001	Vasco	111
2	Teixeira	222	23	002	João	111
3	Alberto	111	23	003	Altino	222
4	Sousa	222	23	004	Ricardo	333

SQL LMD:
SELECT Estudante.nome, Professor.nome, Professor.classe
FROM Estudante, Professor
WHERE Estudante.turma = Professor.classe

OUTPUT		
ESTUDANTE NOME	PROFESSOR NOME	PROFESSOR CLASSE
Carlos	Vasco	111
Alberto	Vasco	111
Carlos	João	111
Alberto	João	111
Teixeira	Altino	222
Sousa	Altino	222

NATURAL JOIN EXEMPLO						
TABELA ITEM				TABELA EMPRESA		
ITEM_ID	ITEM_NAME	ITEM_UNIT	EMPRESA_ID	EMPRESA_ID	EMPRESA_NOME	EMPRESA_CIDADE
1	Francesinha	Dose	003	001	Petisqueira Moura	Lousada
2	Pizza Queijo	Dose	002	002	Pizzaria Ricardo	Lousada
3	Cachorro	Dose	002	003	Francesinha	Lousada
4	Polvo	Dose	004	004	Cedro	Penafiel
5	Pizza Atum	Dose	002	005	Bel Bel	Felgueiras
6	Lombo	Dose	001			
7	Costelas	Dose				

SQL LMD:
SELECT *
FROM Item
NATURAL JOIN Empresa;

OUTPUT					
Empresa_ID	Item_ID	Item_Nome	Item_Unit	Empresa_Nome	Empresa_Cidade
003	1	Francesinha	Dose	Francesinha	Lousada
002	2	Pizza Queijo	Dose	Pizzaria Ricardo	Lousada
002	3	Cachorro	Dose	Pizzaria Ricardo	Lousada
004	4	Polvo	Dose	Cedro	Penafiel
002	5	Pizza Atum	Dose	Pizzaria Ricardo	Lousada
001	6	Lombo	Dose	Petisqueira Moura	Lousada

OUTER JOIN EXEMPLO						
TABELA COMIDA				TABELA EMPRESA		
Item_ID	Item_Nome	Item_Unit	Empresa_ID	Empresa_ID	Empresa_Nome	Empresa_Cidade
1	Francesinha	Dose	003	001	Petisqueira Moura	Lousada
2	Pizza Queijo	Dose	002	002	Pizzaria Ricardo	Lousada
3	Cachorro	Dose	002	003	Francesinha	Lousada
4	Polvo	Dose	004	004	Cedro	Penafiel
5	Pizza Atum	Dose	002	005	Bel Bel	Felgueiras
6	Lombo	Dose	001			
7	Costelas	Dose				

SELECT Empresa.Empresa_Nome, Empresa.Empresa_ID, Comida.Empresa_ID,
 Comida.Item_Nome, Comida.Item_Unit
FROM Empresa, Comida
WHERE Empresa.Empresa_ID = Comida.Empresa_ID

OUTPUT				
Empresa_Nome	Empresa_ID	Empresa_ID	Item_Nome	Item_Unit
Francesinha	003	003	Francesinha	Dose
Pizzaria Ricardo	002	002	Pizza Queijo	Dose
Pizzaria Ricardo	002	002	Cachorro	Dose
Cedro	004	004	Polvo	Dose
Pizzaria Ricardo	002	002	Pizza Atum	Dose
Petisqueira Moura	001	001	Lombo	Dose
Bel Bel	005			

SEMI JOIN EXEMPLO							
Tabela PROFESSOR			TABELA DEPARTAMENTO		PROFESSOR ⋈ DEPARTAMENTO		
Nome	Prof_ID	Dept	Dept	Gestor	Nome	Prof_ID	Dept
Vasco	001	BD	PS	Maria	João	002	PS
João	002	PS	PS	Emilia	Ricardo	004	POO
Altino	003	BD	POO	José			
Ricardo	004	POO	POO	Luis			

O que são Triggers de bases de dados e para que servem?
Um TRIGGER “dispara” uma ação ou conjunto de ações que devem ser realizadas quando algum evento ocorre na aplicação.
Discuta as vantagens e desvantagens dos Triggers
Nas VANTAGENS podem ser destacadas a eliminação de código redundante, a melhora da integridade de informação, a maior facilidade na alteração das regras de integridade do negócio e uma boa junção com a arquitetura cliente-servidor. Quanto as DESVANTAGENS podem se enunciar o overhead do processador, o possível efeito cascata, a falta de possibilidade de agenda os Triggers, e a diminuição da portabilidade uma vez que geralmente cada SGBD possui uma forma diferente de criar Triggers.
Qual a diferença entre Triggers Before, After, Instead Of.
O TRIGGER BEFORE é executado antes da operação que o ativa, sendo um INSERT, um UPDATE, ou um DELETE, Um TRIGGER AFTER , é feito após o término de uma das operações acima referidas. Quando aos TRIGGERS INSTEAD OF estes são executados no lugar de uma operação SQL “normal”.
O que é uma transação? Dê exemplos.
É algo que acontece na BD podendo ser uma inserção, atualização ou remoção de dados. Estas transações terão de respeitar o princípio COMMIT/ROOLBACK, ou seja, se tudo correr bem, a transação é concluída e é feito o COMMIT da operação, se algo correr mal, entra em ação o mecanismo de ROOLBACK que desfaz tudo o que foi realizado na transação.
Discuta as maneiras pelas quais uma transação pode ser completa.
Uma transação pode ser completada de duas maneiras. Através do COMMIT quando a transação é completa com sucesso e alterações tornam-se permanentes ou através do ROOLBACK quando algo de errado acontece, levando à destruição de tudo o que já foi feita na transação e deixando a BD como estava antes da mesma.
Q1. O que são Cursores SQL?
Q2. O que são Cursores SQL? Qual o propósito da sua utilização?
Uma query pode retornar um numero arbitrário de tuplos. Para que seja possível aceder a cada um dos tuplos, um de cada vez, podem ser utilizados os CURSORES que funcionam como uma espécie de apontador permitindo assim percorrer a tabela tuplo a tuplo.
Descreva como funcionam os mecanismos de controlo de acesso em SQL.
Os MECANISMOS DE CONTRO DE ACESSO EM SQL assentam em identificadores de autorização e privilégios. IDENTIFICADORES DE AUTORIZAÇÃO são fornecidos aos utilizadores de BD pelo Administrador da BD. Cada objeto na BD tem um dono e esse dono pode oferecer ou retirar PRIVILÉGIOS aos utilizadores nos seus objetos, através do GRANT e REVOKE, respetivamente.
Qual a diferença entre procedimento e uma função?
A grande diferença entre um PROCEDIMENTO e uma FUNÇÃO é o retorno, ou seja, enquanto um PROCEDIMENTO não retorna nada no final da execução, a FUNÇÃO retorna sempre algum valor.

O que é uma vista? Quais as diferenças entre uma vista e uma relação base?
<p>VISTAS são relações virtuais que podem não existir fisicamente na BD e podem ser produzidas em tempo real mediante o pedido do utilizador.</p> <p>Uma RELAÇÃO BASE ao contrário de uma VISTA existe fisicamente na BD e podem ser usadas na criação da vista.</p>
Defina vistas e apresente a sua importância.
<p>VISTAS são relações virtuais que podem não existir fisicamente na BD e podem ser produzidas em tempo real mediante o pedido do utilizador.</p> <p>Este mecanismo é importante pois oferece uma forma flexível de segurança, permitindo esconder partes da BD de certos utilizadores para além disso permite aos utilizadores aceder à informação de forma personalizada e permite simplificar operações complexas nas relações base.</p>
Discuta as vantagens e desvantagens das vistas.
<p>Em relação às VANTAGENS é possível enunciar a segurança melhorada, complexidade reduzida e a personalização.</p> <p>Em relação às DESVANTAGENS podem incluir se as restrições nas atualizações, pois uma vista poderá não ser atualizada, as restrições na estrutura uma vez que a sua estrutura é definida no momento da sua criação e para haver alterações é necessário criar uma nova.</p> <p>Por fim pode se apontar o problema da performance caso a vista envolva a junção de várias tabelas, essa junção terá que ser feita sempre que a vista é acedida.</p>
Que restrições são necessárias para garantir que a vista é atualizável?
<p>O SGBD deve ser capaz de conseguir rastrear cada linha e coluna até aos seus equivalentes na tabela de origem.</p> <p>Isso pode ser garantido confirmando que na VISTA não existe GROUP BY ou HAVING que o FROM apenas refere uma tabela, não é especificado DISTINCT, e não há funções de agregação nem subqueries.</p>
O que é a materialização de vistas e as suas vantagens associadas.
<p>Consiste no seu armazenamento numa tabela temporária na BD fazendo com que o acesso a essa vista seja muito mais rápido aumentando a performance do sistema caso essa vista seja muito vezes consultada.</p>
Q1. Descreva como funciona o mecanismo de materialização de vistas.
Q2. Descreva como funciona o mecanismo de resolução de vistas.
<p>Os nomes das colunas da vista na lista SELECT são traduzidos para os nomes das colunas correspondentes na definição da vista.</p> <p>Os nomes das vistas no FROM são substituídos pelas correspondentes da lista do FROM da definição da vista.</p> <p>O where da query do utilizador é combinado com o where da definição da vista usando o AND.</p> <p>As clausulas GROUP BY e HAVING são copiadas da definição da vista.</p> <p>O ORDER BY é copiado da query e o nome da coluna traduzido para o nome da coluna definição da vista.</p> <p>A query final é então executado para devolver o resultado.</p>
Enuncie quais as principais abordagens para elaborar o desenho de uma base de dados com múltiplas vistas de utilizadores
<p>CENTRALIZADA:</p> <p>Requisitos de cada vista de utilizador é agregada numa só coleção de requisitos.</p> <p>É criado um modelo global de dados baseado neste conjunto de requisitos (que representam todas as vistas dos utilizadores).</p> <p>INTEGRAÇÃO DE VISTAS:</p> <p>São utilizados os requisitos de cada vista de utilizador para criar um modelo de dados separado.</p> <p>Ao modelo de dados que representa uma vista de um utilizador chama-se modelo de dados local, e é composto por diagramas e documentação que descreve os requisitos de uma vista particular de um utilizador sobre a base de dados.</p> <p>Os modelos de dados locais são depois fundidos de forma a produzir o modelo de dados global, que representa a vista global de todos os utilizadores sobre a base de dados.</p> <p>COMBINAÇÃO DAS DUAS ANTERIORES</p>

Descreva quais os passos a desenvolver no ciclo de vida de uma aplicação de Bases de Dados.

PLANEAMENTO DA BASE DE DADOS: planear o procedimento dos passos no desenvolvimento do ciclo de vida de uma aplicação de BD atividades, para que estes sejam realizados de forma correta.

DEFINIÇÃO DO SISTEMA: descreve o objetivo e os limites da aplicação de BD e as vistas globais dos utilizadores.

RECOLHA E ANÁLISE DE REQUISITOS: procedimento que recolhe e analisa a informação sobre a organização que será usada na aplicação de BD. Com esta informação identifica se os requisitos dos utilizadores para o novo sistema.

DESENHO DA BD: é a criação do desenho da BD que irá suportar as operações e objetivos da organização.

SELEÇÃO DO SGBD (OPCIONAL): escolha de um SGBD apropriado para suportar uma aplicação de BD.

DESENHO DA APLICAÇÃO: desenho da interface com o utilizador e das aplicações que usam e processam a BD.

PROTOTIPAGEM (OPCIONAL): construção de um modelo de trabalho da aplicação da BD.

IMPLEMENTAÇÃO: Realização física da BD e do desenho das aplicações.

CONVERSÃO E ALIMENTAÇÃO DE DADOS: Transferência de quaisquer dados já existentes para a nova BD e converter quaisquer aplicações para trabalharem com a nova BD.

TESTES: execução da aplicação com o objetivo de detetar erros.

MANUTENÇÃO OPERACIONAL: procedimento que monitoriza e gere o sistema após a instalação.

Descreva como são usadas as técnicas de descoberta de factos no decorrer dos diferentes estágios do ciclo de vida de uma aplicação de bases de dados.

As técnicas de descoberta de factos que são usadas no decorrer dos diferentes estágios do ciclo de vida de uma aplicação de BD são:

- **A ANÁLISE DA DOCUMENTAÇÃO** que é produzida, tentando perceber os fluxos existentes e o significado dos termos e como estes devem ser representados na base de dados.
- **A ENTREVISTA** cara-a-cara com os funcionários permite descobrir, verificar, esclarecer fatos e gera entusiasmo ao envolver os utilizadores finais. Ainda serve para identificar os requisitos e recolher ideais de melhorias.
- **A OBSERVAÇÃO DO FUNCIONAMENTO DA ORGANIZAÇÃO**, esta é uma técnica eficiente para perceber o sistema, sendo útil quando a validade de determinados dados é posta em causa ou quando a complexidade de certos aspetos do sistema não permite uma explicação clara pelos utilizadores finais.
- **A PESQUISA**, serve para investigar sobre o problema tentando encontrar em fóruns ou outros lugares por soluções já existentes de forma a resolver o problema.

Identifique quais os principais passos a seguir no Desenho Conceptual e Lógico de uma base de dados.

DESENHO CONCEPTUAL:

Construir um modelo conceptual de dados para cada vista de utilizador

Passo 1.1: identificar tipos de entidades

Passo 1.2: identificar tipos de relacionamentos

Passo 1.3: identificar e associar atributos aos tipos de entidades ou relacionamentos

Passo 1.4: determinar os domínios dos atributos

Passo 1.5: determinar os atributos das chaves primárias e candidatas

Passo 1.6: considere o uso de conceitos de modelação avançada (Passo opcional)

Passo 1.7: verificar a redundância do modelo

Passo 1.8: validar o modelo conceptual local com as transações dos utilizadores

Passo 1.9: rever o modelo conceptual de dados com o utilizador

DESENHO LÓGICO:

Construir e validar o modelo lógico de dados de cada vista

Passo 2.1: remover componentes não compatíveis com o modelo relacional (Passo opcional)

Passo 2.2: obter as relações para o modelo lógico de dados

Passo 2.3: validar as relações usando a normalização

Passo 2.4: validar as relações com as transações de utilizadores

Passo 2.5: definir restrições de integridade

Passo 2.6: rever o modelo lógico de dados com o utilizador

Identifique quais os principais fatores críticos de sucesso no desenho de uma base de dados.

- *Trabalhar interactivamente o mais possível com os utilizadores.*
- *Seguir uma metodologia estruturada durante o processo de modelação de dados.*
- *Incorporar as considerações estruturais e de integridade nos modelos de dados.*
- *Reunir a conceptualização, normalização e a validação de transações na metodologia de modelação de dados.*
- *Usar diagramas para representar o mais possível dos modelos de dados.*
- *Usar a Linguagem de Desenho de Bases de Dados (LDBD) para representar semântica de dados adicionais.*
- *Construir um dicionário de dados para acrescentar aos diagramas de modelos de dados.*
- *Estar preparado para repetir passos.*

Explique as diferenças entre o processo de especialização e de generalização no contexto da modelação ER.

A **ESPECIALIZAÇÃO** é o processo de definir as subclasses de uma superclasse com base numa determinada característica.

Por exemplo: SECRETÁRIA, ENGENHEIRO e TÉCNICO especializam EMPREGADO com base no tipo de trabalho.

A **GENERALIZAÇÃO** é o processo funcionalmente inverso da especialização no qual se identificam as características comuns que passarão a caracterizar uma nova superclasse que generaliza as entidades-tipo originais.

POR EXEMPLO:

CARRO (Matrícula, Numero de Lugares, Velocidade Máxima, Preço)

CAMIÃO (Matrícula, Número de Eixos, Tonelagem, Preço)

Descreva o que representam os atributos num diagrama ER e dê exemplos de atributos simples, compostos, multi-valor e derivados.

Num diagrama ER os atributos representam a propriedade de uma entidade ou de um tipo de relação.

- **SIMPLES:** número de cartão de cidadão.
- **COMPOSTO:** o atributo endereço pode ser decomposto em morada, cidade, código postal.
- **MULTI-VALOR:** pode tomar um ou mais valores para cada entidade, por exemplo o atributo grau académico pode ser: licenciado, mestre, doutorado.
- **DERIVADO:** pode ser determinado a partir de outro atributo. Por exemplo o atributo idade pode ser calculado a partir da data de nascimento.

No contexto do modelo relacional de bases de dados, quais os objetivos da normalização de dados? De que forma o processo de normalização poderá afetar, posteriormente, o desempenho da respetiva implementação?

O **OBJETIVO DA NORMALIZAÇÃO** é analisar uma relação com base na sua chave primária e nas dependências funcionais entre atributos, assim sendo consoante a normalização avança as relações tornam-se mais fortes no formato e também menos vulneráveis a anomalias de atualização.

Enuncie as definições de cada Forma Normal.

FORMA NÃO NORMALIZADA: uma tabela que contém um ou mais grupos repetidos.

PRIMEIRA FORMA NORMAL: uma relação em que a intersecção entre uma linha e uma coluna contenha um e um só valor.

SEGUNDA FORMA NORMAL: uma relação que está na 1FN e todos os atributos não pertencentes à chave primária são totalmente dependentes de qualquer chave candidata.

TERCEIRA FORMA NORMAL: uma relação que está na 1FN e na 2FN e na qual nenhum atributo não pertencente à chave primária depende de qualquer chave candidata.

Em que circunstâncias devemos usar desnormalização? Dê um exemplo para suportar a sua resposta.

É o processo de tentar otimizar o desempenho de leitura/consulta a uma BD, adicionando dados redundantes. Por exemplo: uma tabela que armazena registos de publicações de um blog beneficia uma vez que cada publicação foi escrita uma vez e raramente será editada, mas estará constantemente a ser lida.

Descreva a arquitetura de referência para SGBDs Distribuídos em comparação com a arquitetura ANSI/SPARC.

Dada a diversidade, não existe uma arquitetura equivalente à arquitetura de três níveis ANSI/SPARC, assim sendo a arquitetura de referência consiste num conjunto de esquemas globais externos, num esquema global conceptual, num esquema de fragmentação e de alocação e num conjunto de esquemas para cada SGBD local em conformidade com os três níveis ANSI/SPARC. Alguns níveis podem não aparecer, dependendo dos níveis de transparência suportados.

Uma das estratégias usadas para a alocação dos dados nos SGBDs Distribuídos é a Fragmentação. Descreva quais as razões principais que proporcionam a sua utilização e quais as desvantagens.

As principais razões para a utilização da Fragmentação são:

- **USO:** Aplicações trabalham com vistas e não com relações completas.
- **EFICIÊNCIA:** Dados são armazenados o mais perto possível de onde são mais frequentemente utilizados. Dados que não são necessários às aplicações locais não são armazenados.
- **PARALELISMO:** com fragmentos como unidades de distribuição, as transações podem ser divididas em várias subqueries que trabalham com fragmentos.
- **SEGURANÇA:** dados não necessários às aplicações locais não são armazenados, como tal também não são acedidos por utilizadores não autorizados.

DESVANTAGENS: Performance e integridade.

Descreva quais as quatro estratégias alternativas para a alocação de dados num SGBD Distribuído.

As quatro estratégias alternativas para a alocação de dados num SGBD Distribuído são:

CENTRALIZADA: consiste numa só BD e um SGBD armazenados num site com utilizadores distribuídos por uma rede.

PARTICIONADA (OU FRAGMENTADA): BD particionada em fragmentos independentes, cada um atribuído a um site.

REPLICAÇÃO COMPLETA: consiste em manter uma cópia completa da BD em cada site.

REPLICAÇÃO SELETIVA: combinação de particionamento, replicação e centralização.

Descreva quais as principais arquiteturas para SGBDs Paralelos.

As principais arquiteturas para os SGBDs Paralelos são:

- **MEMÓRIA PARTILHADA:** processadores partilham uma memória comum;
- **DISCO PARTILHADO:** processadores partilham um disco comum;
- **NADA PARTILHADO:** processadores não partilham memória comum nem disco comum;

Explique por palavras suas as razões/condicionantes que proporcionaram o desenvolvimento de SGBD de Objetos.

Abstração, encapsulamento, information hiding.

Objetos e atributos.

Identidade do Objeto.

Métodos e mensagens.

Classes, subclasses, superclasses, e herança.

Overloading.

Polimorfismo e dynamic binding.

Explique porquê que as fraquezas do modelo relacional e dos SGBD Relacionais fazem com que estes sejam inapropriados para aplicações de bases de dados avançadas.

Pois as fraquezas fazem com que haja uma pobre representação de entidades do “Mundo-Real” e de um pobre suporte a restrições de integridade e de negócio, assim como apenas tem uma construção para representar dados e relacionamentos entre dados: a relação.

Os SGBD Relacionais ainda apresentam apenas um conjunto fixo de operações que não pode ser expandido e é extremamente difícil de criar queries recursivas.

Represente a arquitetura típica de um Data Warehouse, descrevendo cada uma das suas componentes.

- | | |
|----------------------------------|--------------------------------------|
| • Operational data | • Detailed Data |
| • Operational data store | • Lightly and Highly Summarized Data |
| • ETL Manager (aka Load Manager) | • Archive/Backup Data |
| • Warehouse Manager | • Metadata |
| • Query Manager | • End-User Access Tools |

Descreva as atividades associadas a cada um dos cinco principais fluxos de dados de um Data Warehouse.

INFLOW: processos associados à extração, limpeza e carregamento dos dados dos sistemas fonte para o data warehouse.

UPFLOW: processos associados à adição de valor aos dados do warehouse através da sumarização, agregação e distribuição dos dados.

DOWNFLOW: processos associados ao arquivamento e backup dos dados do warehouse.

OUTFLOW: processos associados a disponibilização dos dados aos utilizadores finais.

METAFLOW: processos associados à gestão dos metadados.

Descreva os principais benefícios e problemas associados aos Data Warehouses.

Benefícios:

- Grande potencial do retorno sobre o investimento, Vantagem competitiva, Incremento de produtividade dos decision-makers

Problemas:

- Subestimar os recursos necessários ao carregamento dos dados
- Problemas escondidos nos sistemas fonte
- Dados necessários não capturados
- Crescimento dos pedidos dos utilizadores finais
- Homogeneização dos Dados
- Precisa de grandes recursos
- Dados proprietários
- Manutenção Elevada
- Projetos de Longa Duração
- Complexidade da Integração


Quais as diferenças entre um Data Mart e um Data Warehouse. Identifique quais as razões principais para o desenvolvimento de um Data Mart.

Um **DATA MART** é um subconjunto de um **DATA WAREHOUSE** que suporta os requisitos de um determinado departamento ou função de negócio, já um **DATA WAREHOUSE** é uma coleção de dados orientada a assuntos, integrada, variável no tempo e não-volátil em suporte ao processo de tomada de decisão da administração

AS RAZÕES para o desenvolvimento de um **DATA MART** são:

- Dar aos utilizadores acesso aos dados que precisam de analisar mais frequentemente.
- Providenciar os dados que coincidam com a vista coletiva de um grupo de utilizadores num departamento ou área funcional.
- Melhorar o tempo de resposta aos utilizadores finais através da redução do volume de dados que precisam de aceder.
- Disponibilizar os dados devidamente estruturados como foram especificados nos requisitos das ferramentas dos utilizadores finais.
- Construir um **DATA MART** é mais simples que construir um data warehouse empresarial.
- O custo de implementação de um **DATA MART** é normalmente inferior ao custo necessário para a criação de um data warehouse.
- Potenciais utilizadores de um **DATA MART** são mais facilmente definidos e podem ser mais facilmente descobertos de forma a obter maior suporte à criação de um projeto de um **DATA MART** ao invés de um projeto de um data warehouse empresarial.

NORMALIZAÇÃO

 <small>escola superior de tecnologia e gestão de engenharias</small>	Pauta de Prova de Avaliação		Ano Lectivo: 2003/2004	
	CODIGO CURSO 1234	CURSO Engenharia Informática	CODIGO DOCENTE 9876	DOCENTE Vasco Caio Santos
	CODIGO DISCIPLINA 4321	DISCIPLINA Bases de Dados	ANO 3.º Ano	
	CODIGO TIPO PROVA 1	TIPO DE PROVA 1ª Frequência		

Nº Mat.	Nome do Aluno	Nota	Obs.
1234	JOSÉ MANUEL SANTOS	10,0	c)
4313	MARIA LUISA BARBOSA	15,1	a), c)

No final:

DATA
HORARIO DE CONSULTA DA PROVA DE AVALIAÇÃO

a) CONTACTAR DOCENTE b) ADMITIDO A EXAME FINAL c) ADMITIDO A 2ª FREQUÊNCIA d) DISPENSA DE EXAME FINAL e) NÃO COMPARECEU	f) DESISTIU g) APROVADO h) REPROVADO i) DISPENSADO DA PARTE TEÓRICA DO EXAME FINAL j) DISPENSADO DA PARTE PRÁTICA DO EXAME FINAL
---	--

Obs: Apenas pode haver uma pauta por ano lectivo, disciplina e tipo de prova.

FORMA NÃO NORMALIZADA: uma tabela que contém um ou mais grupos repetidos.

Tabela Pauta: Ano Lectivo, Código Curso, Curso, Código Docente, Docente, Código Disciplina, Disciplina, Ano, Código Tipo Prova, Tipo de Prova, Nº Matricula, Nome do Aluno, Nota, Obs, Descrição Obs, Data, Horário Consulta

PRIMEIRA FORMA NORMAL: uma relação em que a intersecção entre uma linha e uma coluna contenha um e um só valor.

Chave Primária: Ano Letivo, Código Disciplina, Código Tipo Prova, Nº Matricula, Obs

Dependências Parciais:

Código Disciplina -> Código Curso, Curso, Disciplina, Ano

Código Tipo Prova -> Tipo de Prova

Nº Matricula -> Nome do Aluno

Obs -> Descrição Obs

Ano Letivo, Código Disciplina -> Código Docente, Docente

Ano Letivo, Código Disciplina, Código Tipo Prova -> Data, Horário Consulta

Ano Letivo, Código Disciplina, Código Tipo Prova, Nº Matricula -> Nota

SEGUNDA FORMA NORMAL: uma relação que está na 1FN e todos os atributos não pertencentes à chave primária são totalmente dependentes de qualquer chave candidata.

Disciplina (Código Disciplina, Disciplina, Código Curso, Curso, Ano)

Tipo Prova (Código Tipo Prova, Tipo de Prova)

Aluno (Nº Matricula, Nome do Aluno)

Observações (Obs, Descrição Obs)

Regente (Ano Letivo, Código Disciplina, Código Docente, Docente)

Consulta (Ano Letivo, Código Disciplina, Código Tipo Prova, Data, Horário Consulta)

Nota (Ano Letivo, Código Disciplina, Código Tipo Prova, Nº Matricula, Nota)

Pauta (Ano Letivo, Código Disciplina, Código Tipo Prova, Nº Matricula, Obs)

Dependências Transitivas:

Código Curso -> Curso

Código Docente -> Docente

TERCEIRA FORMA NORMAL: uma relação que está na 1FN e na 2FN e na qual nenhum atributo não pertencente à chave primária depende de qualquer chave candidata.

Disciplina (Código Disciplina, Disciplina, Código Curso, Curso, Ano)

TipoProva (Código Tipo Prova, Tipo de Prova)

Aluno (Nº Matricula, Nome do Aluno)

Observações (Obs, Descrição Obs)

Regente (Ano Letivo, Código Disciplina, Código Docente, Docente)

Consulta (Ano Letivo, Código Disciplina, Código Tipo Prova, Data, Horário Consulta)

Nota (Ano Letivo, Código Disciplina, Código Tipo Prova, Nº Matricula, Nota)

Pauta (Ano Letivo, Código Disciplina, Código Tipo Prova, Nº Matricula, Obs)

Curso (Código Curso, Curso)

Docente (Código Docente, Docente)