



1. O nível de granularidade adotado numa tabela de factos é unicamente relevante segundo as perspetivas técnica e de projeto. **Falso - Para além dessas 2, também de negócio**
2. Abstraindo de todos os outros fatores que possam condicionar a escolha do nível de granularidade, a melhor opção consiste em adotar o nível de granularidade mais elementar. **Verdadeiro - Independentemente do contexto de negócio, com os dados no nível mais elementar possível, qualquer consulta será possível ser realizada**
3. A utilização de chaves de substituição nas dimensões, em detrimento das chaves dos sistemas operacionais, justifica-se unicamente por questões de performance. **Falso - Para além de performance, tb queremos manter a integridade referencial**
4. A existência de atributos do tipo *EffectiveDate* e *ExpiredDate* ou *IsCurrent* é dispensável na implementação de um mecanismo de *Slowly Changing Dimension* (SCD) – Tipo 2. **Falso - Apenas o IsCurrent é dispensável**
5. A implementação de um mecanismo de *Slowly Changing Dimension* (SCD) – Tipo 2 para armazenamento do histórico das alterações que ocorrem aos atributos de uma dimensão, pode ser plenamente alcançada utilizando unicamente um atributo que indica se é o registo mais atual ou não (*isCurrent*). **Falso - Sem o effectiveDate e o ExpiredDate o conceito de histórico não existe**
6. Considerando que um atributo vai sofrer inúmeras alterações ao longo do tempo, um mecanismo de controlo de alterações de *Slowly Changing Dimension* (SCD) – Tipo 3 é o mais adequado. **Falso - O tipo 3 não é adequado para várias alterações de um atributo**
7. Uma dimensão *Junk* é utilizada em armazéns de dados para armazenar as combinações possíveis de um conjunto de atributos demográficos, o que permite a sua remoção de uma dimensão. **Falso - É uma minidimensão em vez de uma junk**
8. Uma mini-dimensão é utilizada em armazéns de dados para armazenar conjuntos de atributos do tipo textual e do tipo *flag* (booleano), o que permite a sua remoção da tabela de factos. **Falso - É uma junk e não uma minidimensão**
9. Uma tabela de factos é sempre composta pela chave primária, formada pelo conjunto ou subconjunto de chaves estrangeiras, por um conjunto de medidas (factos) cuja análise é relevante para o negócio e, eventualmente, uma ou mais dimensões degeneradas. **Falso - Pode ser factless, logo não tem medidas**
10. Uma tabela de factos pode armazenar três tipos de medidas: aditivas; semi-aditivas; e, não aditivas. **Falso - Não armazena medidas não-aditivas**
11. Na extração de dados estática (a partir dos sistemas operacionais), a única estratégia disponível consiste em utilizar força bruta, ou seja, comparar os atributos dos registos do sistema operacional com os atributos dos registos no armazém de dados. **Falso - Existe outra estratégia, CRC também é uma abordagem**

12. *Kimball (Bus architecture)* e *Inmon (CIF architecture)* concordam que o maior poder dos dados se encontra no nível mais atómico. **Verdade - Porque permite qualquer consulta posterior**
13. Sabe-se que a generalidade das análises/consultas de dados efetuadas num armazém de dados têm sempre o aspeto temporal (data e/ou tempo) presente. Assim, na definição da chave primária da tabela de factos e, consequentemente, do respetivo índice, há que ter este aspeto em consideração. **Verdade, na criação de índices o atributo data deve estar na primeira posição**
14. Numa tabela de factos existe sempre somente um único índice, i.e., o que está inerente à sua chave primária.
15. No processo de carregamento de dados, recorrendo ao componente *Slowly Changing Dimension (Integration Services Project)*, a *business key* corresponde necessariamente a um único atributo.
16. No SQL Server é possível definir-se um índice do tipo *Bitmap* para atributos de dimensões que apresentem baixa cardinalidade (i.e., reduzido n.º de valores distintos).