Questão 1:

```sql
/*1) Listar os seguinte dados das tabelas: invoices (Invoiceid, invoiceDate), invoice_items (Invoiceitemid,
unitprice), total_da_fatura (resultado do somatório de todos os unitprice).
*/
SELECT
    invoiceId,
    invoiceDate,
    InvoiceLineId AS InvoiceItemId,
    UnitPrice,
    SUM(UnitPrice) AS total_da_fatura
FROM invoices
    JOIN invoice_items USING(InvoiceId)

GROUP BY InvoiceId;
```

	InvoiceId	InvoiceDate	InvoiceItemId	UnitPrice	total_da_fatura
1	1	2009-01-01 00:00:00	1	0.99	1.98
2	2	2009-01-02 00:00:00	3	0.99	3.96
3	3	2009-01-03 00:00:00	7	0.99	5.94
4	4	2009-01-06 00:00:00	13	0.99	8.91
5	5	2009-01-11 00:00:00	22	0.99	13.86
6	6	2009-01-19 00:00:00	36	0.99	0.99

```
Execução finalizada sem erros.
Resultado: 412 linhas retornadas em 8 ms
Na linha 1:
/*1) Listar os seguinte dados das tabelas: invoices (Invoiceid, invoiceDate), invoice_items (Invoiceitemid,
unitprice), total_da_fatura (resultado do somatório de todos os unitprice).
*/
SELECT
            invoiceId,
            invoiceDate,
            InvoiceLineId AS InvoiceItemId,
            UnitPrice,
            SUM(UnitPrice) AS total_da_fatura
FROM invoices
            JOIN invoice_items USING(InvoiceId)

GROUP BY InvoiceId;
```

Questão 2:

```sql
/*2) Listar os seguinte dados das tabelas: tracks (Trackid, Name, Albumid), albums (Title, Artistid),
artists(Name).*/

SELECT
    TrackId,
    tracks.name AS Track_Name,
    AlbumId,
    Title,
    ArtistId,
    artists.name AS Artist_Name
    FROM albums
JOIN artists USING(ArtistId)
JOIN tracks USING(AlbumId);
```

	TrackId	Track_Name	AlbumId	Title	ArtistId	Artist_Name
1	1	For Those About To Rock (We Salute …	1	For Those About To Rock We Salute You	1	AC/DC
2	6	Put The Finger On You	1	For Those About To Rock We Salute You	1	AC/DC
3	7	Let's Get It Up	1	For Those About To Rock We Salute You	1	AC/DC
4	8	Inject The Venom	1	For Those About To Rock We Salute You	1	AC/DC
5	9	Snowballed	1	For Those About To Rock We Salute You	1	AC/DC

```
Execução finalizada sem erros.
Resultado: 3503 linhas retornadas em 14 ms
Na linha 1:
/*2) Listar os seguinte dados das tabelas: tracks (Trackid, Name, Albumid), albums (Title, Artistid),
artists(Name).*/

SELECT
            TrackId,
            tracks.name AS Track_Name,
            AlbumId,
            Title,
            ArtistId,
            artists.name AS Artist_Name
            FROM albums
JOIN artists USING(ArtistId)
JOIN tracks USING(AlbumId);
```

Questão 3:

```sql
/*3) Listar os seguintes dados das tabelas: tracks(Trackid, Name, Milliseconds), mediatypes (MediaTypeid,
name), genres (GenreId, name). Selecionar as tracks com milliseconds > 2000000.*/
SELECT
    TrackId,
    tracks.name AS Track_Name,
    Milliseconds,
    MediaTypeId,
    media_types.Name AS media_type_name,
    GenreId,
    genres.name AS genre_Name
FROM tracks
    JOIN media_types USING(MediaTypeId)
    JOIN genres USING(GenreId)
WHERE Milliseconds > 2000000;
```

	TrackId	Track_Name	Milliseconds	MediaTypeId	media_type_name	GenreId	genre_Name
1	2819	Battlestar Galactica: The Story So …	2622250	3	Protected MPEG-4 video file	18	Science Fiction
2	2825	A Measure of Salvation	2563938	3	Protected MPEG-4 video file	18	Science Fiction
3	2826	Hero	2713755	3	Protected MPEG-4 video file	18	Science Fiction
4	2827	Unfinished Business	2622038	3	Protected MPEG-4 video file	18	Science Fiction
5	2828	The Passage	2623875	3	Protected MPEG-4 video file	18	Science Fiction

```
Execução finalizada sem erros.
Resultado: 160 linhas retornadas em 14 ms
Na linha 1:
/*3) Listar os seguintes dados das tabelas: tracks(Trackid, Name, Milliseconds), mediatypes (MediaTypeid,
name), genres (GenreId, name). Selecionar as tracks com milliseconds > 2000000.*/

SELECT
            TrackId,
            tracks.name AS Track_Name,
            Milliseconds,
            MediaTypeId,
            media_types.Name AS media_type_name,
            GenreId,
            genres.name AS genre_Name
FROM tracks
            JOIN media_types USING(MediaTypeId)
            JOIN genres USING(GenreId)
WHERE Milliseconds > 2000000;
```