

- report
 - 过程
 - 1.商家开展“买200赠100”的促销活动，为一次购买某专辑总价格在200元（含）以上的用户的会员卡（字段：**User**表的**UserAdvancePayment**）充值100元。（只充值一次）
 - 2. 新建“专辑销售统计表”**Sales_Statis**，其结构能存放专辑名、销售量和销售额。将各个专辑销售量和销售额的统计结果存入**Sales_Statis**表中
 - 3.创建专辑名为“三十而立”（可自定义其他名称）的歌曲视图**SongsByAlbumName**，该视图包括“歌曲表”中的所有列以及专辑名称，并按歌曲顺序号由小到大排列
 - 4.将此视图中选择任意**SongId**，修改歌曲名称为“我就想改改名字看看能不能成功”，看看是否成功？
 - 5.将此视图中选择任意**SongId**，修改歌曲名称为“我就想改改名字看看能不能成功”，专辑名称修改为“我还想改一下专辑试试”，看看是否成功？
 - 6.对于视图是否能修改成功，相关的因素有哪些？请一一列举并举例说明，附上测试结果
 - 测试用例：
 - 完整代码

report

过程

1.商家开展“买200赠100”的促销活动，为一次购买某专辑总价格在200元（含）以上的用户的会员卡（字段：User**表的**UserAdvancePayment**）充值100元。（只充值一次）**

Database: @localhost > musicshop > 表 > users

	UserAgeRange	UserAddress	UserPostCode	UserPhone	UserEmail	UserRegisterTime	UserAdvancePayment
1	无可奉告	北京市 朝阳区	100024	13600088855	wangtong@cuc.edu.cn	<null>	100.00
2	18~20岁	北京市 海淀区	100098	13583011235	wyh@cuc.edu.cn	<null>	100.00
3	41~50岁	武汉市 东湖区	430077	13064010321	zhangxia@sina.com	<null>	<null>
4	无可奉告	北京市 朝阳区	100024	13600088855	woshintong@cuc.edu.cn	<null>	<null>
5	小于18岁	上海市 徐汇区	200018	13901010486	li@sina.com	<null>	<null>
6	21~30岁	天津市 开发区	300067	13751010461	guojing@sina.com	<null>	<null>

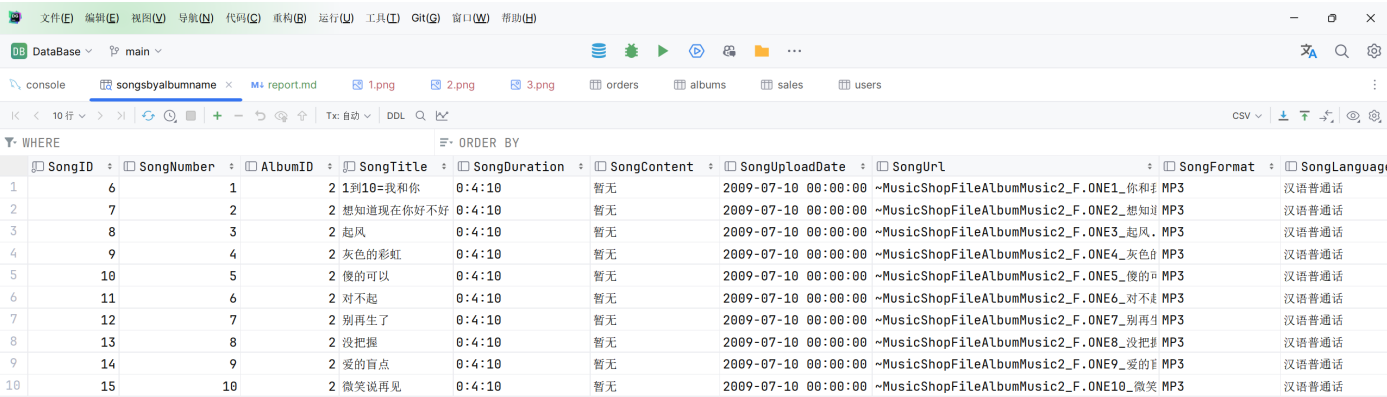
2. 新建“专辑销售统计表”**Sales_Statis**，其结构能存放专辑名、销售量和销售额。将各个专辑销售量和销售额的统计结果存入**Sales_Statis**表中

Database: @localhost > musicshop > 表 > sales_statis

	AlbumName	Quantity	TotalPrice
1	F.ONE	3	255.00
2	世纪好歌	9	630.00
3	三十而立	2	200.00
4	最炫民族风	1	80.00
5	好好好	11	880.00

3. 创建专辑名为“三十而立”（可自定义其他名称）的歌曲视图**SongsByAlbumName**，该视图

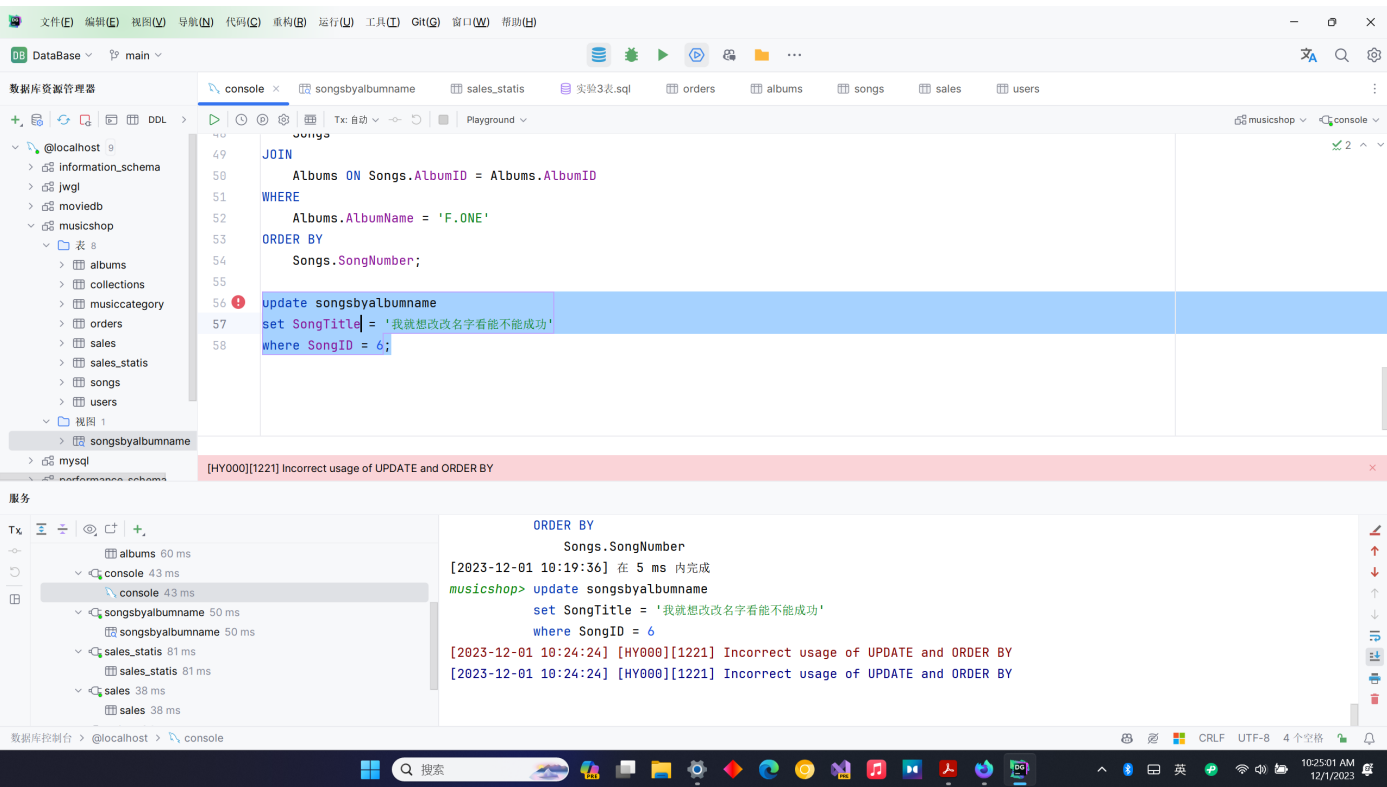
包括“歌曲表”中的所有列以及专辑名称，并按歌曲顺序号由小到大排列



The screenshot shows a database client interface with a table view of 'songsbyalbumname'. The table has columns for SongID, SongNumber, AlbumID, SongTitle, SongDuration, SongContent, SongUploadDate, SongUrl, SongFormat, and SongLanguage. The data is sorted by SongNumber in ascending order.

SongID	SongNumber	AlbumID	SongTitle	SongDuration	SongContent	SongUploadDate	SongUrl	SongFormat	SongLanguage
1	6	1	2 1到10=我和你	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE1_我和你	MP3	汉语普通话
2	7	2	2 想知道现在你好不好	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE2_想知道	MP3	汉语普通话
3	8	3	2 起风	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE3_起风	MP3	汉语普通话
4	9	4	2 灰色的彩虹	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE4_灰色	MP3	汉语普通话
5	10	5	2 傻的可以	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE5_傻的	MP3	汉语普通话
6	11	6	2 对不起	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE6_对不起	MP3	汉语普通话
7	12	7	2 别再生了	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE7_别再生	MP3	汉语普通话
8	13	8	2 没把握	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE8_没把握	MP3	汉语普通话
9	14	9	2 爱的盲点	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE9_爱的	MP3	汉语普通话
10	15	10	2 微笑说再见	0:4:10	暂无	2009-07-10 00:00:00	~MusicShopFileAlbumMusic2_F.ONE10_微笑	MP3	汉语普通话

4.将此视图选择任意SongId，修改歌曲名称为“我就想改改名字看看能不能成功”，看看是否成功？



The screenshot shows a database client interface with a console window displaying an SQL update statement and its execution results. The statement is:

```
update songsbyalbumname
set SongTitle = '我就想改改名字看看能不能成功'
where SongID = 6;
```

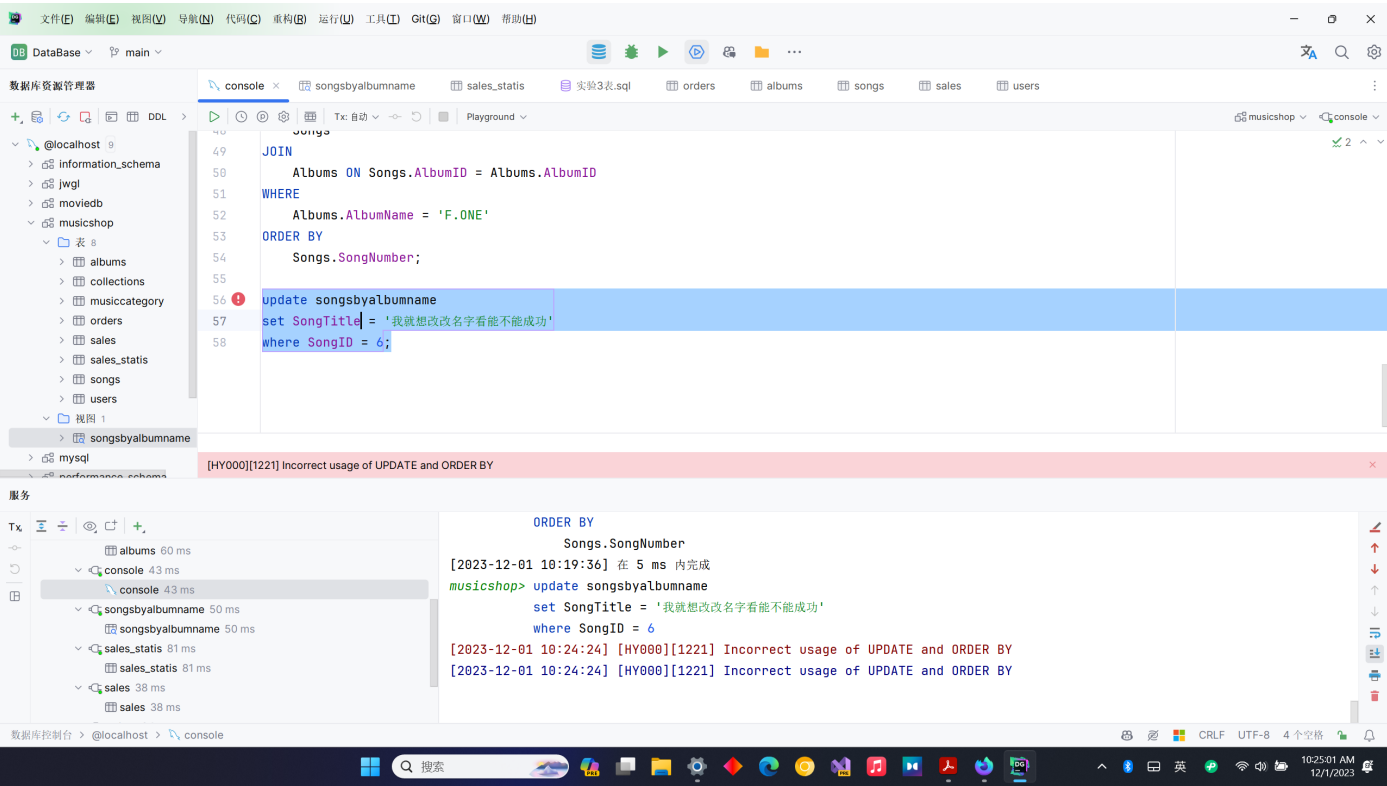
The execution results show an error: [HY000][1221] Incorrect usage of UPDATE and ORDER BY.

The console window also shows the execution of a SELECT statement to verify the update:

```
select * from songsbyalbumname
where SongID = 6;
```

The results of the SELECT statement show the updated song title: '我就想改改名字看看能不能成功'.

5.将此视图中选择任意SongId，修改歌曲名称为“我就想改改名字看看能不能成功”，专辑名称修改为“我还想改一下专辑试试”，看看是否成功？



6.对于视图是否能修改成功，相关的因素有哪些？请一一列举并举例说明，附上测试结果

视图的修改能否成功，主要取决于以下几个因素：

视图的定义：如果视图是由多个表联接而成，或者包含聚合函数、分组、DISTINCT等，那么这个视图就是不可更新的。例如，我们在上面的例子中创建的"SongsByAlbumName"视图就是不可更新的，因为它是由"Songs"表和"Albums"表联接而成的。

基础表的约束：如果基础表有任何阻止更新的约束（如主键、外键、CHECK约束等），那么视图也不能更新。例如，如果"Songs"表的"SongID"是主键，那么我们不能通过视图更改这个字段的值。

视图的权限：如果用户没有对基础表的更新权限，那么他们也不能更新视图。例如，如果用户只有对"SongsByAlbumName"视图的SELECT权限，而没有对"Songs"表和"Albums"表的UPDATE权限，那么他们就不能通过视图更新歌曲名称或专辑名称。

视图的WITH CHECK OPTION：如果视图定义中包含WITH CHECK OPTION，那么只

有当更新后的数据仍满足视图的WHERE条件时，更新才会成功。例如，如果我们在"SongsByAlbumName"视图的定义中添加了WITH CHECK OPTION，并且WHERE条件是"Albums.AlbumName = '三十而立'"，那么只有当更新后的专辑名称仍为"三十而立"时，更新才会成功。

测试用例：

```
-- 创建一个可更新的视图
CREATE VIEW UdatableSongs AS
SELECT SongID, SongTitle
FROM Songs;

-- 尝试更新视图
UPDATE UdatableSongs
SET SongTitle = '新的歌曲名称'
WHERE SongID = 1;

-- 查询视图来确认更改是否成功
SELECT * FROM UdatableSongs WHERE SongID = 1;

-- 创建一个包含WITH CHECK OPTION的视图
CREATE VIEW SongsByAlbumNameWithCheck AS
SELECT Songs.SongID, Songs.SongTitle, Albums.AlbumName
FROM Songs
JOIN Albums ON Songs.AlbumID = Albums.AlbumID
WHERE Albums.AlbumName = '三十而立'
WITH CHECK OPTION;

-- 尝试更新视图
UPDATE SongsByAlbumNameWithCheck
SET SongTitle = '新的歌曲名称', AlbumName = '新的专辑名称'
WHERE SongID = 1;

-- 查询视图来确认更改是否成功
SELECT * FROM SongsByAlbumNameWithCheck WHERE SongID = 1;
```

完整代码

```
-- 创建一个临时表来存储已经充值的用户
create temporary table if not exists temp_1(
    username varchar(50)
);

-- 找出购买专辑总价格在200元（含）以上的订单
insert into temp_1(username)
select distinct orders.UserName
from orders
```

```

where GoodsFee >= 200;

-- 为这些用户的会员卡充值100元
update users
set UserAdvancePayment = UserAdvancePayment + 100
where UserName in (select username from temp_1);

-- 删除临时表
drop table temp_1;


-- 创建“专辑销售统计表”Sales_Statis
CREATE TABLE Sales_Statis(
    AlbumName varchar(64),
    Quantity int,
    TotalPrice numeric(10, 2)
);

-- 存入数据
insert into Sales_Statis(albumname, quantity, totalprice)
select
    Albums.AlbumName,
    SUM(Sales.Quantity),
    SUM(Sales.Quantity * Albums.AlbumMemberPrice)
from
    sales
    JOIN
    Albums ON Sales.AlbumID = Albums.AlbumID
group by
    Albums.AlbumName;


CREATE VIEW SongsByAlbumName AS
SELECT
    Songs.*,
    Albums.AlbumName
FROM
    Songs
JOIN
    Albums ON Songs.AlbumID = Albums.AlbumID
WHERE
    Albums.AlbumName = 'F.ONE'
ORDER BY
    Songs.SongNumber;


update songsbyalbumname
set SongTitle = '我就想改改名字看能不能成功'
where SongID = 6;

update songsbyalbumname
set SongTitle = '我就想改改名字看能不能成功',
    AlbumName = '我还想改改专辑试试'
where SongID = 6;

```

```
-- 创建一个可更新的视图
CREATE VIEW UpdatableSongs AS
SELECT SongID, SongTitle
FROM Songs;

-- 尝试更新视图
UPDATE UpdatableSongs
SET SongTitle = '新的歌曲名称'
WHERE SongID = 1;

-- 查询视图来确认更改是否成功
SELECT * FROM UpdatableSongs WHERE SongID = 1;

-- 创建一个包含WITH CHECK OPTION的视图
CREATE VIEW SongsByAlbumNameWithCheck AS
SELECT Songs.SongID, Songs.SongTitle, Albums.AlbumName
FROM Songs
JOIN Albums ON Songs.AlbumID = Albums.AlbumID
WHERE Albums.AlbumName = 'Eye Fever演唱会09'
WITH CHECK OPTION;

-- 尝试更新视图
UPDATE SongsByAlbumNameWithCheck
SET SongTitle = '新的歌曲名称', AlbumName = '新的专辑名称'
WHERE SongID = 1;

-- 查询视图来确认更改是否成功
SELECT * FROM SongsByAlbumNameWithCheck WHERE SongID = 1;
```