Oracle® Tutor™

Section 4 数据仓库注意事项和答案

1.文档

第 4 场景：

14231:
p470 create external table
p490 partition table
14215:
p187 sqlldr

2.考点

1.快速刷新物化视图
2.可更新的物化视图
3.SQL*Loader
4.DataPump
5.备份
6.确保库 open,能本地连接/网络连接

3.难点

1）TRAILING NULLCOLS 意味着如果行的位数不够，都以 null 值插入
对应于外部表的语法是：MISSING FIELD VALUES ARE NULL

举例如下：
sqlldr_file.dat 的内容：

```
P,James,31,
P,Thomas,22,
E,Pat,38,93645,1122,Engineering,
P,Bill,19,
P,Scott,55,
S,Judy,45,27316,English,
S,Karen,34,80356,History,
E,Karen,61,90056,1323,Manufacturing,
S,Pat,29,98625,Spanish,
S,Cody,22,99743,Math,
P,Ted,43,
E,Judy,44,87616,1544,Accounting,
E,Bob,50,63421,1314,Shipping,
S,Bob,32,67420,Psychology,
E,Cody,33,25143,1002,Human Resources,
```

建立一个控制文件 sqlldr.ctl

```
load data
infile 'sqlldr_file'
append into table sqlldr_test
TRAILING NULLCOLS
(col1 CHAR TERMINATED BY ",",
col2 CHAR TERMINATED BY ",",
col3 INTEGER EXTERNAL TERMINATED BY ",",
col4 INTEGER EXTERNAL TERMINATED BY ",",
col5 INTEGER EXTERNAL TERMINATED BY ",",
col6 CHAR TERMINATED BY ","
)
```

注意因为有几行的数据只有 3 列 columns，有些有 6 列；对于不到 6 列的行，后面的几列必须制定 tailing nullcols，用 null 来补齐。

2）请看“建立一个能快速刷新的物化视图，且能去除表中的重复行“ 这一块。
如果没有 count(*)，是不支持 update 和 delete 的，是个只支持 insert 的 insert-only materialized view。
Oracle 文档中有明确说明：
If the materialized view has one of the following, then fast refresh is supported only on conventional DML inserts and direct loads.
* Materialized views with MIN or MAX aggregates
* Materialized views which have SUM(expr) but no COUNT(expr)
* Materialized views without COUNT(*)

3）如果考 mv 的排错，建议用 dbms_advisor.tune_mview 和 dbms_mview.explain_mview 来排错，比如 DISTINCT 操作完全可以转换为 GROUP BY：
如果只是包含一个 DISTINCT 操作，那么完全可以转换为 GROUP BY 语句：
SQL> CREATE TABLE T AS

```
2 SELECT *
3 FROM DBA_OBJECTS;
Table created.
SQL> CREATE MATERIALIZED VIEW LOG ON T
2 WITH ROWID, SEQUENCE (OWNER, OBJECT_TYPE)
3 INCLUDING NEW VALUES;
Materialized view log created.
SQL> CREATE MATERIALIZED VIEW MV_T REFRESH FAST
2 AS SELECT DISTINCT OWNER, OBJECT_TYPE
3 FROM T;
FROM T
*
ERROR at line 3:
ORA-12015: cannot create a fast refresh materialized view from a complex query

SQL> CREATE MATERIALIZED VIEW MV_T_COMPLETE
2 AS SELECT DISTINCT OWNER, OBJECT_TYPE
3 FROM T;
Materialized view created.
```

显然包含 DISTINCT 的物化视图只能完全刷新而不能快速刷新，不过这种物化视图可以很轻松的改成下面的写法：

```
SQL> CREATE MATERIALIZED VIEW MV_T REFRESH FAST
2 AS SELECT OWNER, OBJECT_TYPE, COUNT(*) COUNT
3 FROM T
4 GROUP BY OWNER, OBJECT_TYPE;
Materialized view created.
```

物化视图使用 GROUP BY 的写法和使用 DISTINCT 的写法是等价的，而且可以使用快速刷新。唯一不同是多了一个 COUNT(*)的字段，而一般情况下，物化视图的额外列没有多大影响。如果基表不包括 UPDATE 和 DELETE，只有 INSERT 操作，那么物化视图中可以去掉 COUNT(*)列。

```
SQL> CREATE MATERIALIZED VIEW MV_T1 REFRESH FAST
2 AS SELECT OWNER, OBJECT_TYPE
3 FROM T
4 GROUP BY OWNER, OBJECT_TYPE;
Materialized view created.
SQL> DELETE T WHERE ROWNUM = 1;
1 row deleted.
SQL> EXEC DBMS_MVIEW.REFRESH('MV_T')
PL/SQL procedure successfully completed.
SQL> EXEC DBMS_MVIEW.REFRESH('MV_T1')
BEGIN DBMS_MVIEW.REFRESH('MV_T1'); END;
*
ERROR at line 1:
ORA-32314: REFRESH FAST of "TEST"."MV_T1" unsupported after deletes/updates
ORA-06512: at "SYS.DBMS_SNAPSHOT", line 2254
ORA-06512: at "SYS.DBMS_SNAPSHOT", line 2460
ORA-06512: at "SYS.DBMS_SNAPSHOT", line 2429
ORA-06512: at line 1
```

可以看到如果不包括 COUNT(*)，则物化视图不支持 UPDATE 和 DELETE 语句的快速刷新。

4）外部表的并行度问题：

2）创建外部表。

```
create table x_1
organization external
( type oracle_datapump
default directory dir_dmp
location ('X_1.dmp','X_2.dmp')
)
PARALLEL
as
select * from dba_objects;
```

4）查询新建的外部表。

```
select *from x_1;
```

注意：

parallel 默认是 2

parallel<文件数，结果文件数以 parallel 为准

parallel>文件数，结果以指定文件数为准

(查看文件方法 strings X_1.dmp > 1.txt)

4.考题

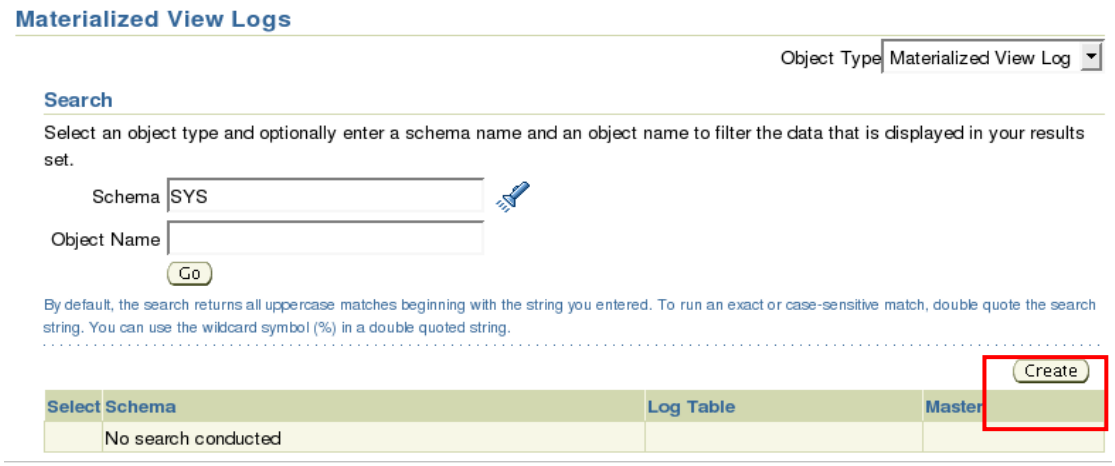
1. Fast Refreshable Materialized View

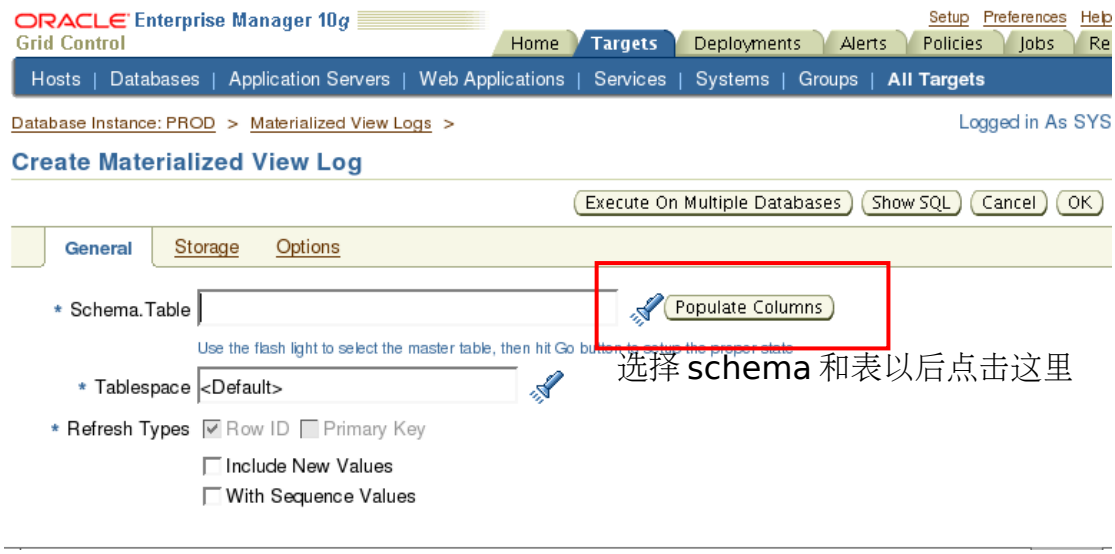
1.1 Using the query found in the mview1.txt text file.create a fast refreshable materialized view named PROD_MV in the SH schema.

```
SELECT time_id,prod_subcategory,SUM(unit_cost),
COUNT(unit_cost),COUNT(*)
FROM costs c,products p
where c.prod_id=p.prod_id
GROUP BY time_id,prod_subcategory;
```

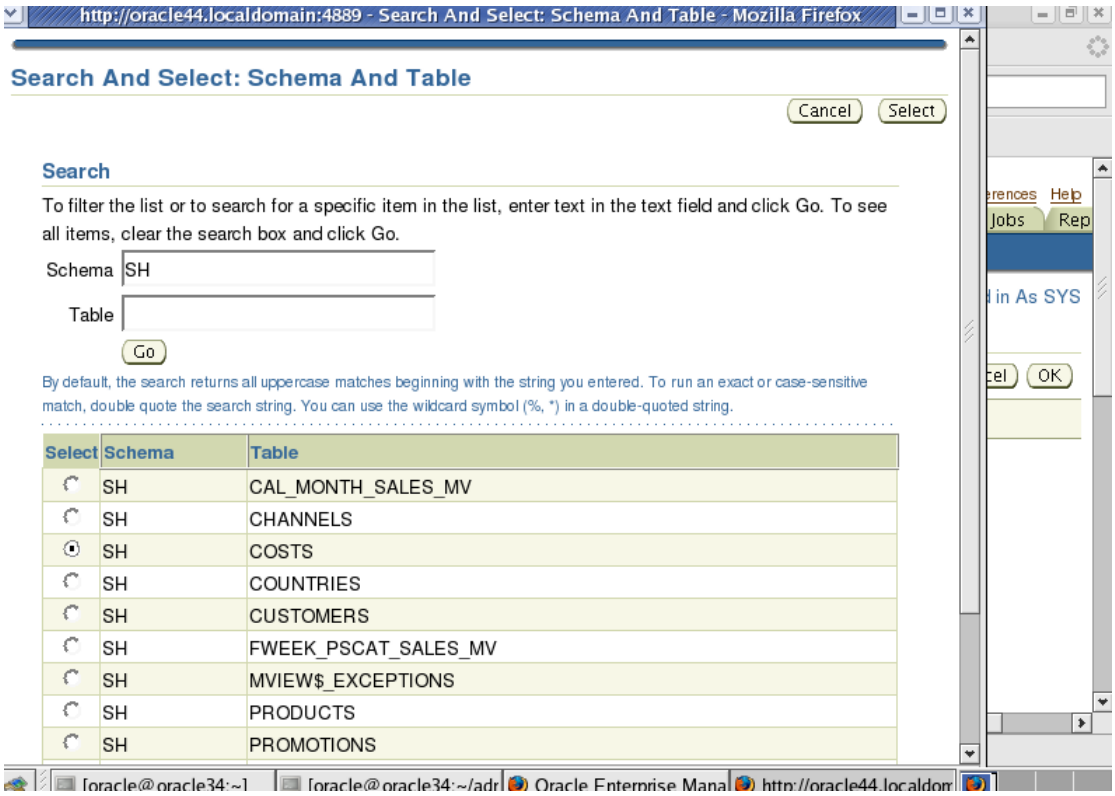
如果使用图形界面做：

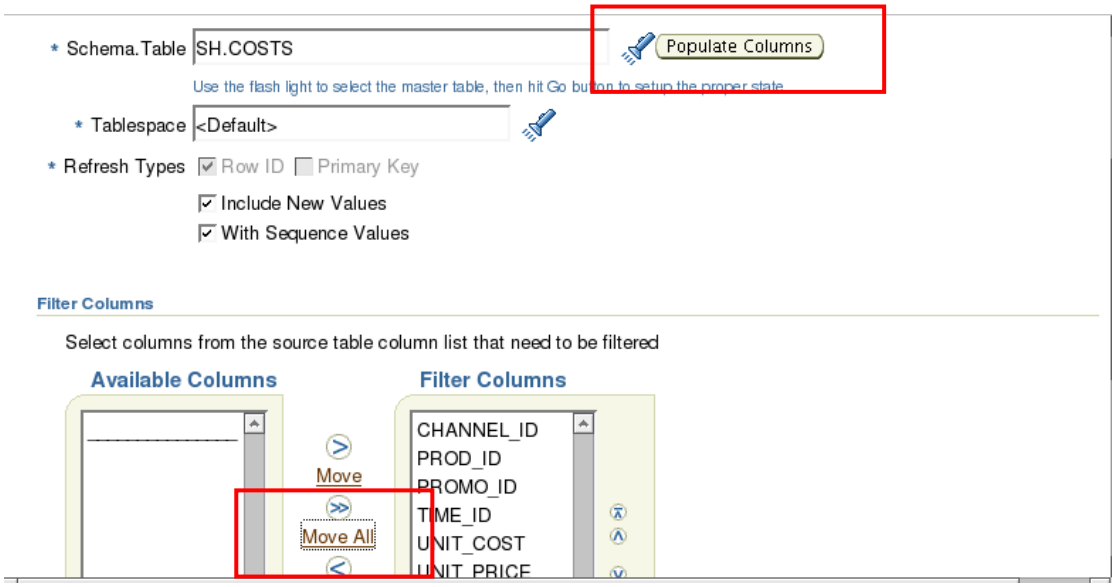
1.先创建表 costs、products 的 Materialized View Log





选择 schema 和表以后点击这里



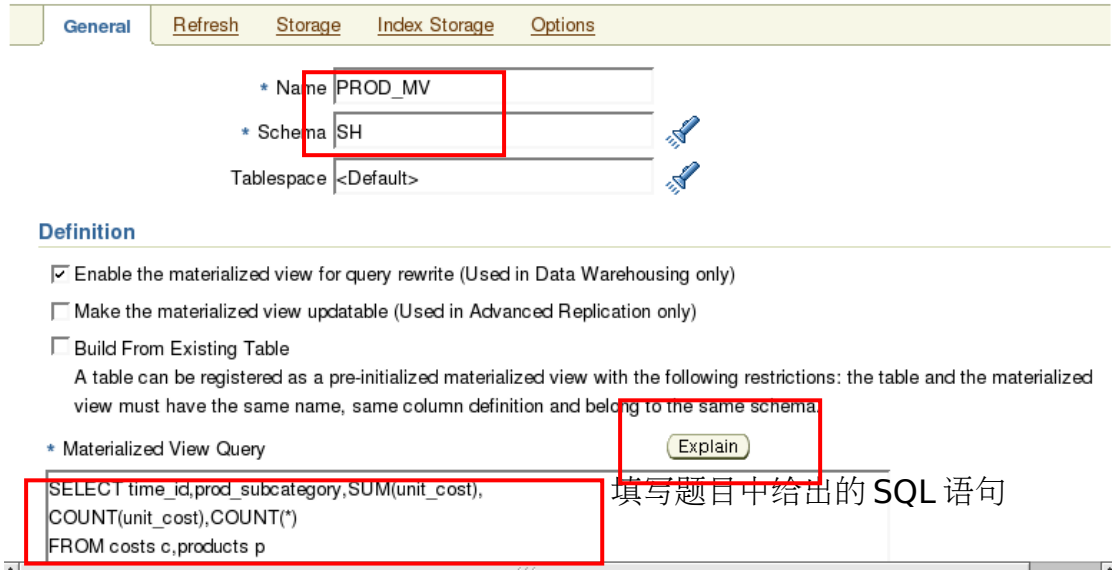


如果使用命令做：

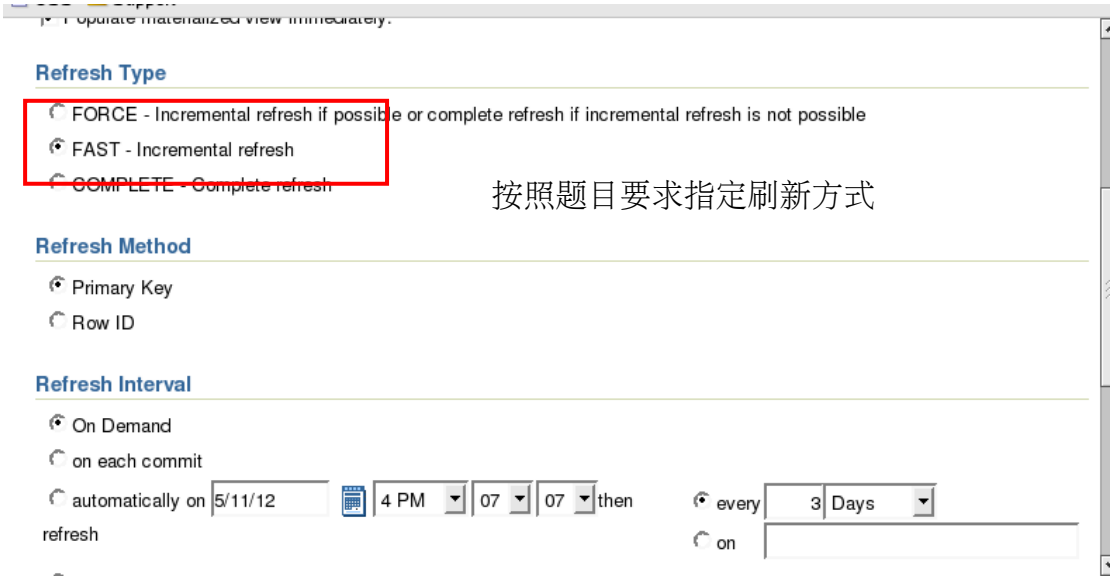
```
CREATE MATERIALIZED VIEW LOG ON costs
WITH SEQUENCE
(prod_id, unit_cost, time_id, channel_id, promo_id, unit_price)
INCLUDING NEW VALUES;
CREATE MATERIALIZED VIEW LOG ON products
WITH SEQUENCE, ROWID,PRIMARY
(/*所有字段*/)
INCLUDING NEW VALUES;
```

2.创建 Materialized View

如果使用图形界面做：



填写题目中给出的 SQL 语句



按照题目要求指定刷新方式

如果使用命令做：

```
CREATE MATERIALIZED VIEW PROD_MV
BUILD IMMEDIATE
REFRESH FAST
ENABLE QUERY REWRITE
AS SELECT time_id,prod_subcategory,SUM(unit_cost),
COUNT(unit_cost),COUNT(*)
FROM costs c.products p
where c.prod_id=p.prod_id
GROUP BY time_id,prod_subcategory;
```

2.Creating an Updatable Materialized View

2.1. Using the HR.EMPLOYEES table in the PROD database. create an updatable materialized view in the EMREP database named EMP_UPD_MV consisting of the following columns: EMPLOYEE_ID,FIRST_NAME,LAST_NAME,PHONE_NUMBER,SALARY.

1.创建 database link

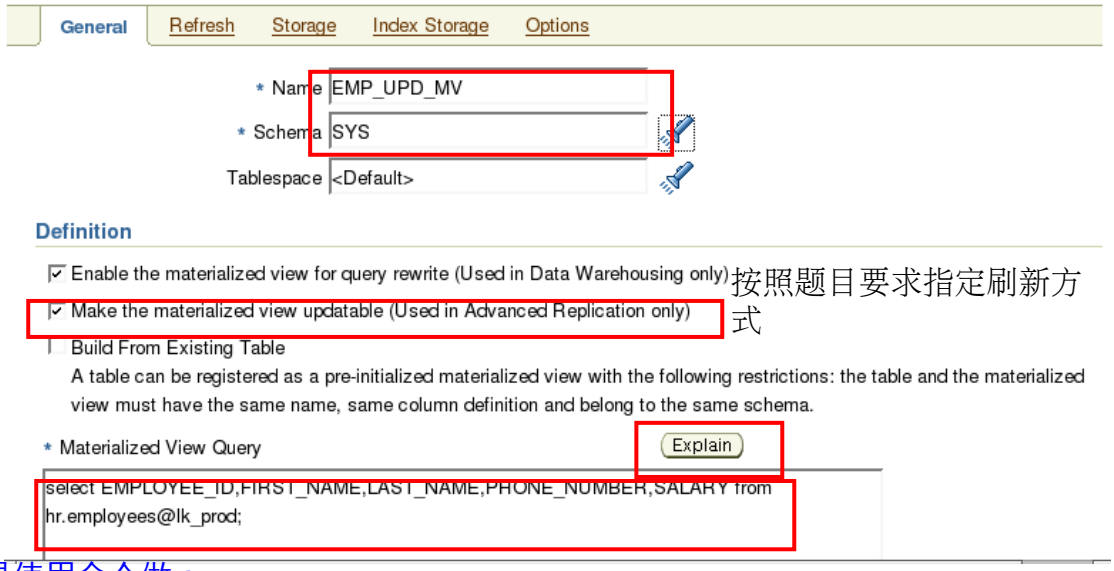
首先，在两个数据库上设置初始化参数 global_names 的值为 false。（目的是为 database link 命名时，不必与远程数据库的 global database name 相同）

```
create database link lk_prod connect to hr identified by hr using 'prod';
```

2.验证 database link 是否可用

```
select EMPLOYEE_ID,FIRST_NAME,LAST_NAME,PHONE_NUMBER,SALARY from hr.employees@lk_prod;
```

3.创建 Updatable Materialized View



按照题目要求指定刷新方式

如果使用命令做：

```
CREATE MATERIALIZED VIEW EMP_UPD_MV
BUILD IMMEDIATE
for update
ENABLE QUERY REWRITE
as
SELECT EMPLOYEE_ID，FIRST_NAME，LAST_NAME，PHONE_NUMBER，SALARY from
EMPLOYEES@dl_prod;
```

3.Oracle_Loader External Tables

3.1. In the scripts directory. you will find prod_master.dat and prod_master.ctl.Using the information found in these files.create and external table named PROD_MASTER in the SH schema of the PROD database.

记着设置“badfile”参数，防止第一次没有导进去，可以方便第二次导和问题诊断

```
[oracle@red1 script]$ cat prod_master.dat
1 1 tom 1
2 2 rose 2
3 1 jone 2
4 3 jack 3
5 2 jacky 4
[oracle@red1 script]$ cat prod_master.ctl
load data
infile '/home/oracle/script/prod_master.dat'
into table sh.exm1 fields terminated by whitespace
TRAILING NULLCOLS
(emp_no,dept_no,name,num)
```

创建外部表的语法：

```
CREATE TABLE PROD_MASTER
(
  "EMP_NO" NUMBER,
  "DEPT_NO" NUMBER,
  "NAME" VARCHAR2(300),
  "NUM" NUMBER
)
```

```
ORGANIZATION external
(
  TYPE oracle_loader
  DEFAULT DIRECTORY mydir1
  ACCESS PARAMETERS
  (
    RECORDS DELIMITED BY NEWLINE CHARACTERSET US7ASCII
    BADFILE mydir1:'prod_master.bad'
    LOGFILE 'prod_master.log_xt'
    READSIZE 1048576
    FIELDS TERMINATED BY WHITESPACE LDRTRIM
    MISSING FIELD VALUES ARE NULL
    REJECT ROWS WITH ALL NULL FIELDS
    (
      "EMP_NO" CHAR(255)
        TERMINATED BY WHITESPACE,
      "DEPT_NO" CHAR(255)
        TERMINATED BY WHITESPACE,
      "NAME" CHAR(255)
        TERMINATED BY WHITESPACE,
      "NUM" CHAR(255)
        TERMINATED BY WHITESPACE
    )
  )
  location
  (
    'prod_master.dat'
  )
)REJECT LIMIT UNLIMITED
;
```

4. Oracle_Datapump External Table

4.1. Create an external table called COUNTRIES_EXT in the PROD database owned by SH. containing the data from the COUNTRY_ID,COUNTRY_NAME,and COUNTRY_REGION columns of the SH.COUNTRIES table.

```
1.创建目录
Create directory dir1 as '/home/oracle/';
Grant read,write on directory dr1 to sh;
2.创建外部表
CREATE TABLE sh.COUNTRIES_EXT
    ORGANIZATION EXTERNAL
    (
      TYPE ORACLE_DATAPUMP
      DEFAULT DIRECTORY dir1
      LOCATION ('COUNTRIES_EXT.dat')
```

```
)
PARALLEL
REJECT LIMIT UNLIMITED
 as
 select COUNTRY_ID,COUNTRY_NAME,COUNTRY_REGION from SH.COUNTRIES;
```

4.2. Create another external table called COUNTRIES_EXT in the EMREP database owned by SYSTEM.The source of the data is the external file(s) created in the previous step.

```
 CREATE TABLE sh.COUNTRIES_EXT
 (
      COUNTRY_ID NUMBER,
      COUNTRY_NAME VARCHAR2(40),
      COUNTRY_REGION VARCHAR2(20)
 )
ORGANIZATION EXTERNAL
(
  TYPE ORACLE_DATAPUMP
  DEFAULT DIRECTORY dir1
  LOCATION ('COUNTRIES_EXT.dat')
)
PARALLEL
REJECT LIMIT UNLIMITED;
```