جامعة دمشق
كلية الهندسة المعلوماتية
السنة الخامسة
قسم البرمجيات

مشروع قواعد البيانات المتقدمة

إعداد الطلاب :

فادي أبو ليل **حسين كحول**

ريم الزيفة

طارق أحمد **لجين المرعي**

1- Oracle:

- الطلب الأول :

تعريف TableSpace باسم homeworkts حجمه 400MB مؤلف من اربع ملفات معطيات datafile :

```
Enter user-name: sys
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> create tablespace homeworkts datafile 'DF1.dbf' size 100 M,
  2  'DF2.dbf' size 100 M,
  3  'DF3.dbf' size 100 M,
  4  'DF4.dbf' size 100 M;

Tablespace created.
```

- الطلب الثاني :

تعريف profile باسم homeworkpf يسمح لمستخدم واحد الاتصال بقاعدة المعطيات حيث تكون مدة الاتصال الفعال ساعة والغير فعال عشر دقائق وصلاحية كلمة المرور سبعة أيام :

```
CREATE PROFILE homeworkpf LIMIT
SESSIONS_PER_USER 1
CONNECT_TIME 60
IDLE_TIME 10
PASSWORD_LIFE_TIME 7;
```

Script Output

Task completed in 0.251 seconds

```
*Action:   Use a different name for the new tablespace

Profile HOMEWORKPF created.
```

- **الطلب الثالث :**

تعريف حساب user في قاعدة المعطيات باسم homeworku مرتبط ب homeworkts ومنحه homeworkpf :

```
CREATE USER homeworku IDENTIFIED BY pass
DEFAULT TABLESPACE homeworkts
PROFILE homeworkpf;
```

Script Output | Task completed in 0.249 seconds

```
Profile HOMEWORKPF created.

User HOMEWORKU created.
```

- **الطلب الرابع :**

إعطاء كافة الصلاحيات للمستخدم homeworku من خلال Roles واحدة وأن يكون له صلاحيات القراءة من جدول ال departement وتعديل وحذف من جدول ال employees :

اولاً: انشاء roles واعطائها الصلاحيات :

```
CREATE ROLE first_role NOT IDENTIFIED;
GRANT ALL PRIVILEGES TO first_role;
GRANT SELECT ON hr.departments TO first_role;
GRANT DELETE,UPDATE ON hr.employees TO first_role;
```

Script Output | Task completed in 0.402 seconds

```
Role FIRST_ROLE created.

Grant succeeded.

Grant succeeded.

Grant succeeded.
```

ثانياَ : منح ال roles للمستخدم homeworku :

```
GRANT first_role TO homeworku;
```

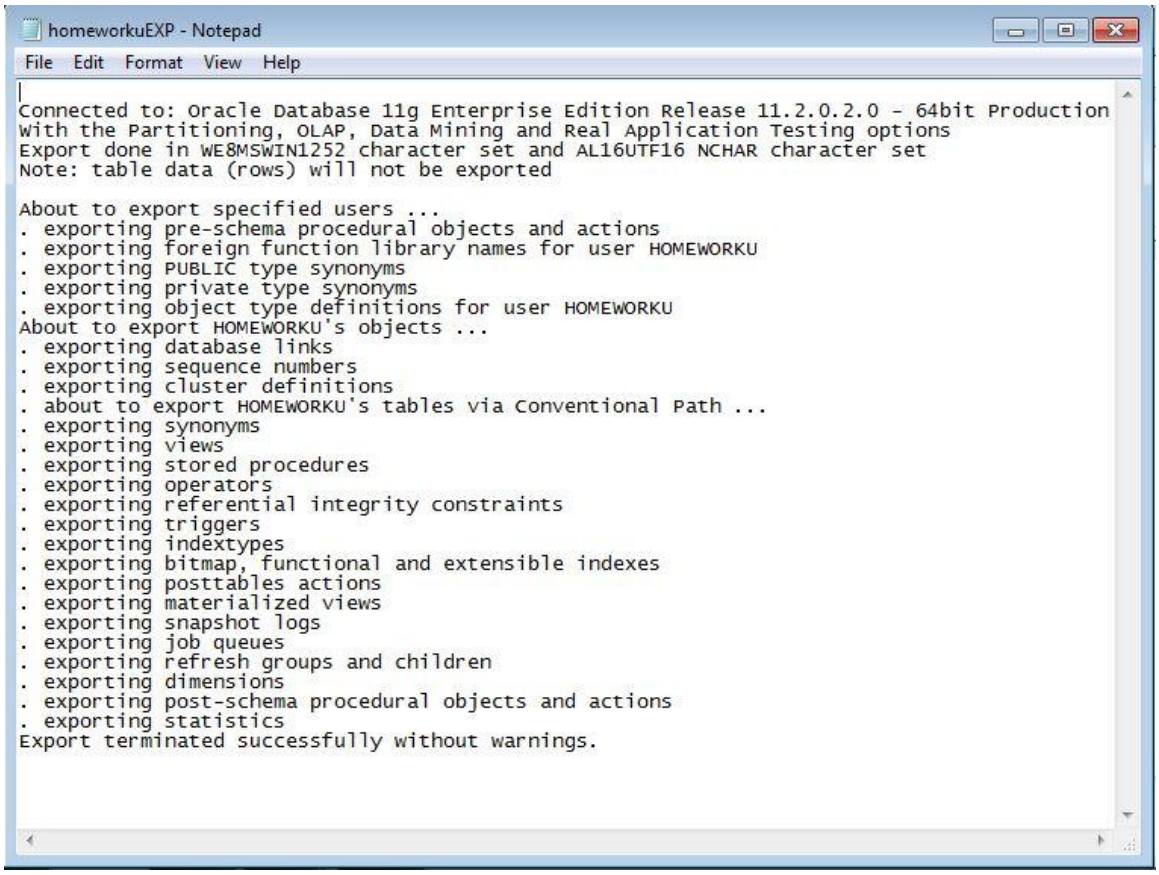
Script Output | Task completed in 0.838 seconds

```
Grant succeeded.
```

- الطلب الخامس :

اجراء نسخة احتياطية للحساب homeworku دون أخذ بيانات الجداول وعرض ملف ال log :

```
C:\Windows\System32\cmd.exe

C:\app\Ali\product\11.2.0\dbhome_1>EXP.EXE USERID='sys/passw0rd as sysdba' OWNER
="homeworku" FILE=c:\backup\homeworku.DMP COMPRESS=Y GRANTS=Y INDEXES=Y LOG=c:\b
ackup\homeworkuEXP.LOG ROWS=N CONSTRAINTS=Y

Export: Release 11.2.0.2.0 - Production on Tue Dec 19 10:09:41 2023

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights reserved.


Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
Note: table data (rows) will not be exported

About to export specified users ...
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user HOMEWORKU
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user HOMEWORKU
About to export HOMEWORKU's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export HOMEWORKU's tables via Conventional Path ...
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully without warnings.
```


homeworkuEXP - Notepad
File Edit Format View Help
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
Note: table data (rows) will not be exported
About to export specified users ...
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user HOMEWORKU
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user HOMEWORKU
About to export HOMEWORKU's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export HOMEWORKU's tables via Conventional Path ...
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully without warnings.

2-PL_SQL :

- الطلب الأول :

إضافة الأفلام تتم فقط الخميس بين الساعة السادسة والثامنة صباحا :

```
CREATE OR REPLACE TRIGGER add_films_checkTime
BEFORE INSERT ON film
DECLARE
 current_day VARCHAR2(10);
 current_hour NUMBER;
BEGIN
 current_day := TO_CHAR(SYSDATE, 'DAY');
 current_hour := TO_NUMBER(TO_CHAR(SYSDATE, 'HH24'));

 IF current_day != 'THURSDAY' AND current_hour NOT BETWEEN 6 AND 8 THEN
   RAISE_APPLICATION_ERROR(-20500, 'insert only allowed between 6 and 8 on Thursday (-_-)');
 END IF;
END;/
```

Script Output

Task completed in 0.128 seconds

```
Trigger ADD_FILMS_CHECKTIME compiled
```

```
INSERT INTO film (id,languageID,title,descripion,release_year,rental_duration,
rental_rate,length,replacement_cost,rating,special_features,fulltext)
VALUES (1,1,'hi','desc',2000,199,5,2,7,8,'hello','fadi');
```

Script Output

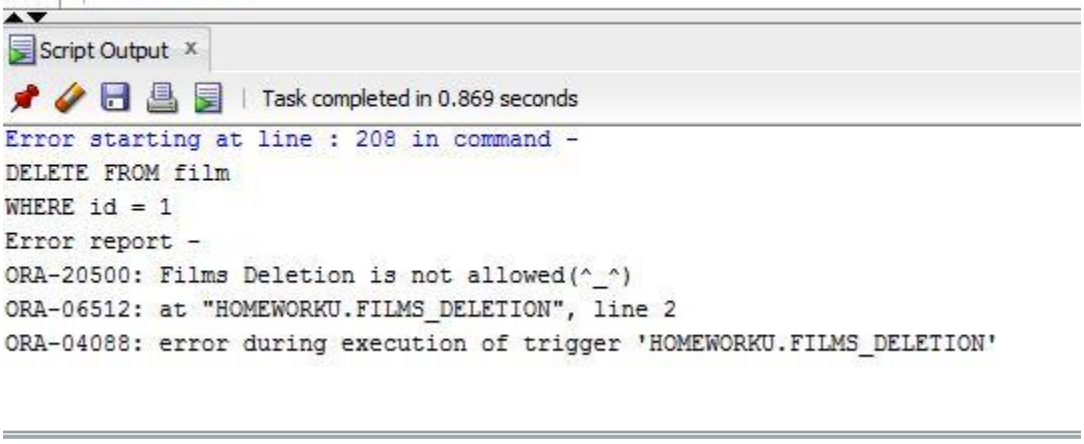
Task completed in 0.128 seconds

```
Error starting at line : 199 in command -
INSERT INTO film (id,languageID,title,descripion,release_year,rental_duration,
rental_rate,length,replacement_cost,rating,special_features,fulltext)
VALUES (1,1,'hi','desc',2000,199,5,2,7,8,'hello','fadi')
Error report -
ORA-20500: insert only allowed between 6 and 8 on Thursday (-_-)
ORA-06512: at "HOMEWORKU.ADD_FILMS_CHECKTIME", line 9
ORA-04088: error during execution of trigger 'HOMEWORKU.ADD_FILMS_CHECKTIME'
```

منع حذف أي فيلم:

```
CREATE OR REPLACE TRIGGER films_deletion
BEFORE DELETE ON film
BEGIN
  RAISE_APPLICATION_ERROR(-20500, 'Films Deletion is not allowed(^_^)');
END;
```


```
DELETE FROM film
WHERE id = 1;
```



- الطلب الثاني :

قيمة الايجار مساوية لكلفة الايجار في جدول الأفلام كل أيام الأسبوع باستثناء يومي السبت والأحد تضاف اليها 15% :

```
CREATE OR REPLACE TRIGGER check_rental_day
BEFORE INSERT OR UPDATE ON payment
FOR EACH ROW
DECLARE
 costT INT;
 current_day VARCHAR2(10);
BEGIN
 current_day := TO_CHAR(SYSDATE, 'DAY');
 SELECT film.replacement_cost INTO costT FROM rental
 JOIN inventory ON rental.inventoryID=inventory.id
 JOIN film ON inventory.filmID=film.id WHERE rental.id = :NEW.rentalID;
 IF current_day IN ('SATURDAY','SUNDAY') THEN
 costT:=costT*1.15;
 END IF;
  :NEW.amount:=costT;
END;
```

- الطلب الثالث :

منع تأجير نفس الفيلم لنفس الزبون أكثر من مرة في الشهر :

```
CREATE OR REPLACE TRIGGER check_customer_monthly_rental
BEFORE INSERT OR UPDATE ON rental
FOR EACH ROW
DECLARE
 countT NUMBER;
BEGIN
 SELECT COUNT(*) INTO countT FROM rental WHERE customerID = :NEW.customerID
 AND inventoryID = :NEW.inventoryID
 AND EXTRACT(MONTH FROM rental_date)=EXTRACT(MONTH FROM SYSDATE);
 IF countT > 0 THEN
 RAISE_APPLICATION_ERROR(-20500, 'You Can Not Rental The Same Film More Than One Time In One Month(+_+)');
 END IF;
END;
```

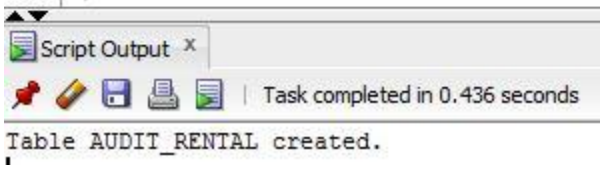

```
SQL> INSERT INTO rental (id,customerID,inventoryID) VALUES (13,1,1);
INSERT INTO rental (id,customerID,inventoryID) VALUES (13,1,1)
            *
ERROR at line 1:
ORA-20500: You Can Not Rental The Same Film More Than One Time In One
Month(+_+)
ORA-06512: at "HOMEWORKU.CHECK_CUSTOMER_MONTHLY_RENTAL", line 8
ORA-04088: error during execution of trigger
'HOMEWORKU.CHECK_CUSTOMER_MONTHLY_RENTAL'
```

- **الطلب الرابع :**

بناء سجل متابعة لعمليات الإضافة والتعديل لجدول الحجوزات:

أولا نقوم ببناء الجدول :

```
CREATE TABLE audit_rental (
  o_staffID INT,
  o_customerID INT,
  o_inventoryID INT,
  o_rental_date TIMESTAMP,
  o_return_date TIMESTAMP,
  o_last_update TIMESTAMP,
  n_staffID INT,
  n_customerID INT,
  n_inventoryID INT,
  n_rental_date TIMESTAMP,
  n_return_date TIMESTAMP,
  n_last_update TIMESTAMP,
  operation_type VARCHAR(20),
  operation_date TIMESTAMP
);
```




```
Table AUDIT_RENTAL created.
```

ثانيا تسجيل العمليات :

:

```
CREATE OR REPLACE TRIGGER audit_rental
AFTER INSERT OR UPDATE OR DELETE ON rental
FOR EACH ROW
DECLARE
 operation_type VARCHAR(20);
BEGIN
 IF INSERTING THEN operation_type := 'I';
 ELSIF UPDATING THEN operation_type := 'U';
 ELSIF DELETING THEN operation_type := 'D';
 END IF;
 INSERT INTO audit_rental VALUES
 (:OLD.staffID,:OLD.customerID,:OLD.inventoryID,:OLD.rental_date,:OLD.return_date,
 :OLD.last_update,:NEW.staffID,:NEW.customerID,:NEW.inventoryID,:NEW.rental_date,
 :NEW.return_date,:NEW.last_update,operation_type,SYSDATE);
END;
```

Script Output x

Task completed in 1.842 seconds

```
Trigger AUDIT_RENTAL compiled
```

- الطلب الخامس :

تسجيل دخول وخروج المستخدمين لقاعدة البيانات :

أولا نقوم ببناء الجدول:

```
CREATE TABLE homeworku.user_connection (
  usr VARCHAR(50),
  operation VARCHAR(20),
  conn_date TIMESTAMP
);
```

Script Output x

Task completed in 0.374 seconds

```
Table HOMEWORKU.USER_CONNECTION created.
```

ثانيا تسجيل عمليات الدخول والخروج:

```
CREATE OR REPLACE TRIGGER user_logon
AFTER LOGON ON DATABASE
BEGIN
 IF user NOT IN ('SYS','SYSTEM') THEN
  INSERT INTO homeworku.user_connection VALUES
  (user,'LOGON',TO_TIMESTAMP(SYSDATE));
 END IF;
END;
```

Script Output | Task completed in 1.394 seconds

```
Trigger USER_LOGON compiled
```

```
CREATE OR REPLACE TRIGGER user_logoff
BEFORE LOGOFF ON DATABASE
BEGIN
 IF user NOT IN ('SYS','SYSTEM') THEN
  INSERT INTO homeworku.user_connection VALUES
  (user,'LOGOFF',TO_TIMESTAMP(SYSDATE));
 END IF;
END;
```

Script Output | Task completed in 0.908 seconds

```
Trigger USER_LOGOFF compiled
```

Connections | HOMEWORKU | USER_CONNECTION

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | De

Sort.. | Filter:

	USR	OPERATION	CONN_DATE
1	HOMEWORKU	LOGON	24-DEC-23 12.00.00.000000000 AM
2	HOMEWORKU	LOGOFF	24-DEC-23 12.00.00.000000000 AM
3	FADI	LOGON	24-DEC-23 12.00.00.000000000 AM
4	FADI	LOGOFF	24-DEC-23 12.00.00.000000000 AM

3- Warehouse:

- الطلب الأول :

نقوم باختيار جدولي rental و payment ليكونا Fact_Table لأنه تتم عليهما اكبر عدد من المناقلات
ويكون لدينا الأبعاد التالية :
-البعد الزمني times :

```
CREATE TABLE times(
  time_id NUMBER(10) CONSTRAINT timeID_pk PRIMARY KEY,
  year INT,
  quarter INT,
  month INT,
  week INT,
  day INT
);
```

Script Output | Task completed in 1.476 seconds

```
Table TIMES created.
```

```
CREATE DIMENSION time_dim
  LEVEL YEAR IS (times.year)
  LEVEL MONTH IS (times.month)
  LEVEL WEEK IS (times.week)
  LEVEL DAY IS (times.day)
  HIERARCHY time_rollup(
    day CHILD OF week CHILD OF month CHILD OF year
  )
;
```

Script Output | Task completed in 0.205 seconds

```
Dimension TIME_DIM created.
```

-بعد الزبائن نقوم ببنائه وتحقيق هرميته :

```
CREATE TABLE customer_dim(
  cust_dim_id NUMBER(10) CONSTRAINT customer_dimID_pk PRIMARY KEY,
  customerID INT , addressID INT,
  address_address VARCHAR(50),
  cityID int,
  city VARCHAR(50),
  countryID INT,
  country VARCHAR(50),
   first_name VARCHAR(255),
  last_name VARCHAR(255),
  email VARCHAR(50),
  active CHAR(1),
  create_date TIMESTAMP,
  last_update TIMESTAMP
);
```

Script Output

Task completed in 0.773 seconds

```
Table CUSTOMER_DIM created.
```

```
CREATE DIMENSION customer_dim
LEVEL addressID IS (customer_dim.addressID)
LEVEL address_address IS (customer_dim.address_address)
LEVEL cityID IS (customer_dim.cityID)
LEVEL city IS (customer_dim.city)
LEVEL countryID IS (customer_dim.countryID)
LEVEL country IS (customer_dim.country)
HIERARCHY address_rollup(
  addressID CHILD OF address_address CHILD OF cityID CHILD OF city CHILD OF countryID CHILD OF country)
;
```

Script Output

Task completed in 0.306 seconds

```
Dimension CUSTOMER_DIM created.
```

- بعد المتاجر مقوم ببنائه وتحقيق هرميته :

```
CREATE TABLE stor_dim(
  stor_dim_id NUMBER(10) CONSTRAINT stor_dimID_pk PRIMARY KEY,
  storID INT,
  addressID INT,
  address VARCHAR(50),
  cityID int,
  city VARCHAR(50),
  countryID INT,
  country VARCHAR(50),
  last_update TIMESTAMP
);
```

Script Output x

Task completed in 0.836 seconds

```
Table STOR_DIM created.
```

```
CREATE DIMENSION stor_dim
  LEVEL addressID IS (stor_dim.addressID)
  LEVEL address IS (stor_dim.address)
  LEVEL cityID IS (stor_dim.cityID)
  LEVEL city IS (stor_dim.city)
  LEVEL countryID IS (stor_dim.countryID)
  LEVEL country IS (stor_dim.country)
  HIERARCHY address_rollup(
    addressID CHILD OF address CHILD OF cityID CHILD OF city
    CHILD OF countryID CHILD OF country)
  ;
```

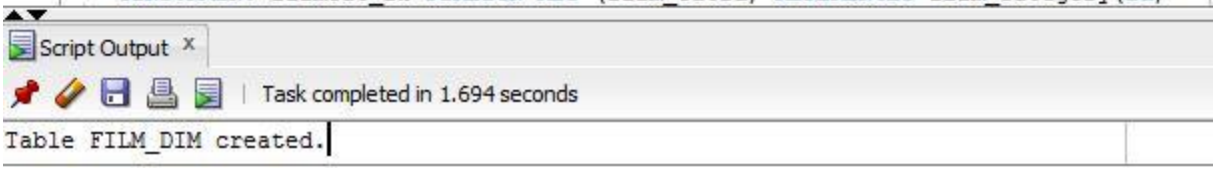
Script Output x

Task completed in 0.424 seconds

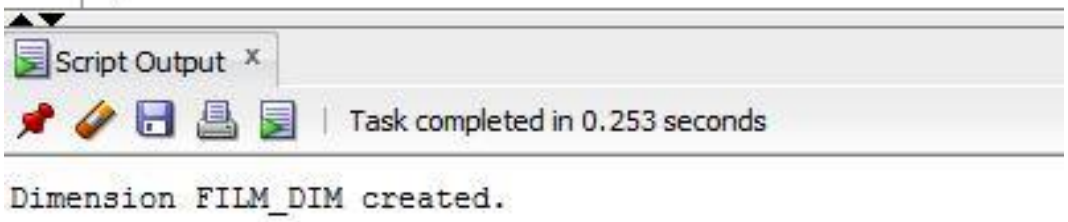
```
Dimension STOR_DIM created.
```

- بعد الأفلام نقوم ببنائه وتحقيق هرميته :

```
CREATE TABLE film_dim (
  film_dim_id NUMBER(10) CONSTRAINT film_dimID_pk PRIMARY KEY,
  filmID INT,
  languageID INT,
  language_name VARCHAR(20),
  film_catId INT,
  title VARCHAR(255),
  descripion VARCHAR(255),
  release_year INT,
  rental_duration INT,
  rental_rate NUMERIC(19,0),
  length INT,
  replacement_cost NUMERIC(19,0),
  rating INT,
  last_update TIMESTAMP,
  special_features VARCHAR(255),
  fulltext VARCHAR(255),
  CONSTRAINT filmCat_fk FOREIGN KEY (film_catId) REFERENCES film_category(id)
```




```
CREATE DIMENSION film_dim
LEVEL languageID IS (film_dim.languageID)
LEVEL language_name IS (film_dim.language_name)
HIERARCHY language_rollup(
  languageID CHILD OF language_name)
;
```



- الطلب الثاني :

نقوم ببناء جدول rental_fact وتجزئته بطريقة range حسب التاريخ لكونه الأكثر استخداما
كما يمكن أن تكون التجزئة مركبة من نوع range-list بحيث نقسم وفق range بحسب التاريخ ونقسم وفق list بحسب المدينة.

```
CREATE TABLE rental_fact(
    rental_fact_id NUMBER(10) CONSTRAINT rental_factID_pk PRIMARY KEY,
    times_id NUMBER,stor_id NUMBER,customer_id NUMBER,film_id NUMBER,
    rental_date TIMESTAMP,return_date TIMESTAMP, amount NUMERIC(19,0),
    payment_date TIMESTAMP,
    CONSTRAINT film_fk FOREIGN KEY (film_id) REFERENCES film_dim(film_dim_id),
    CONSTRAINT time_fk FOREIGN KEY (times_id) REFERENCES times(time_id),
    CONSTRAINT stor_fk FOREIGN KEY (stor_id) REFERENCES stor_dim(stor_dim_id),
    CONSTRAINT customer_fk FOREIGN KEY (customer_id) REFERENCES customer_dim(cust_dim_id))
PARTITION BY RANGE (rental_date)
(
  PARTITION rental_q1 VALUES LESS THAN (TO_TIMESTAMP('01-APR-2023','dd-MON-yyyy')),
  PARTITION rental_q2 VALUES LESS THAN (TO_TIMESTAMP('01-JUL-2023','dd-MON-yyyy')),
  PARTITION rental_q3 VALUES LESS THAN (TO_TIMESTAMP('01-OCT-2023','dd-MON-yyyy')),
  PARTITION rental_q4 VALUES LESS THAN (TO_TIMESTAMP('01-JAN-2024','dd-MON-yyyy'))
);
```

Script Output x

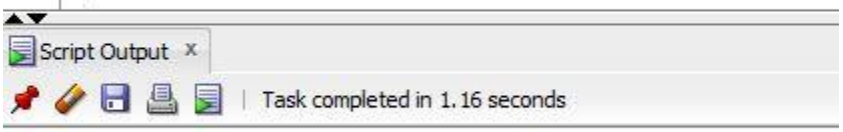
Task completed in 0.752 seconds

```
Table RENTAL_FACT created.
```

- الطلب الثالث :

لنقوم ببناء MATERIALIZED VIEW يتضمن اسم الفيلم واسم الزبون ووقت الحجز يجب أولا ان نقوم ببناء MATERIALIZED VIEW LOG على كل من rental_fact,film_dim,customer_dim :

```
CREATE MATERIALIZED VIEW LOG ON customer_dim
WITH SEQUENCE,PRIMARY KEY,ROWID INCLUDING NEW VALUES;
```

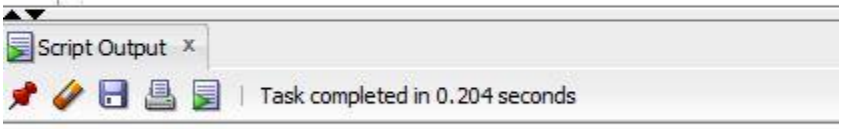



```
Materialized view log CUSTOMER_DIM created.
```

```
CREATE MATERIALIZED VIEW LOG ON film_dim
WITH SEQUENCE,PRIMARY KEY,ROWID INCLUDING NEW VALUES;
```


```
Materialized view log FILM_DIM created.
```

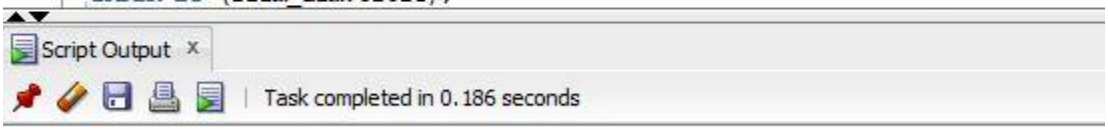
```
CREATE MATERIALIZED VIEW LOG ON rental_fact
WITH SEQUENCE,PRIMARY KEY,ROWID INCLUDING NEW VALUES;
```




```
Materialized view log RENTAL_FACT created.
```

ومن ثم نقوم ببناء ال MATERIALIZED VIEW :

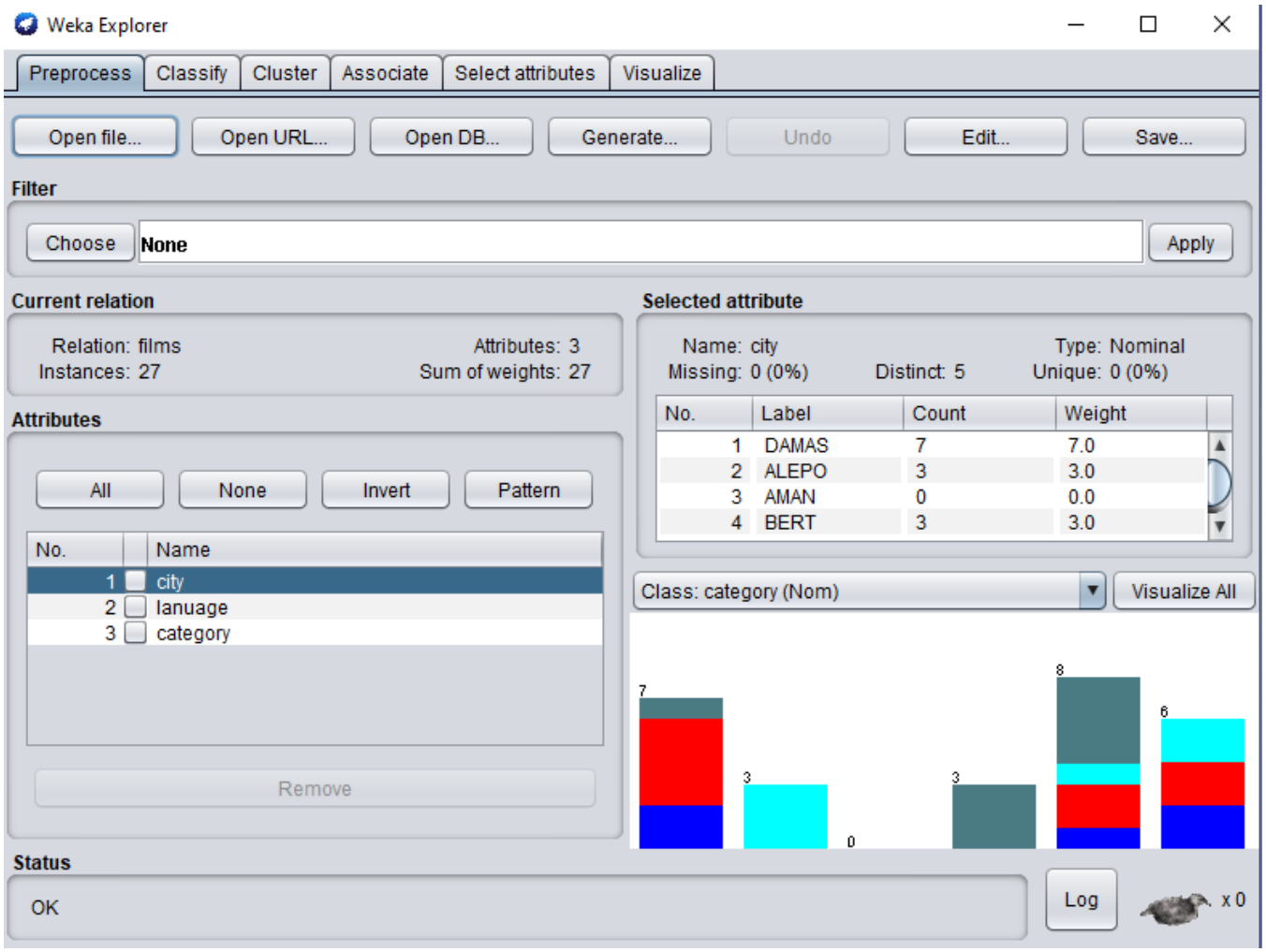
```
CREATE MATERIALIZED VIEW cust_film_data BUILD IMMEDIATE
REFRESH FORCE ON DEMAND AS
SELECT film_dim.title AS film_name ,customer_dim.first_name AS customer,
 rental_fact.rental_date FROM rental_fact
JOIN film_dim ON rental_fact.film_id=film_dim.film_dim_id
JOIN customer_dim ON rental_fact.customer_id=customer_dim.cust_dim_id
ORDER BY (film_dim.title);
```

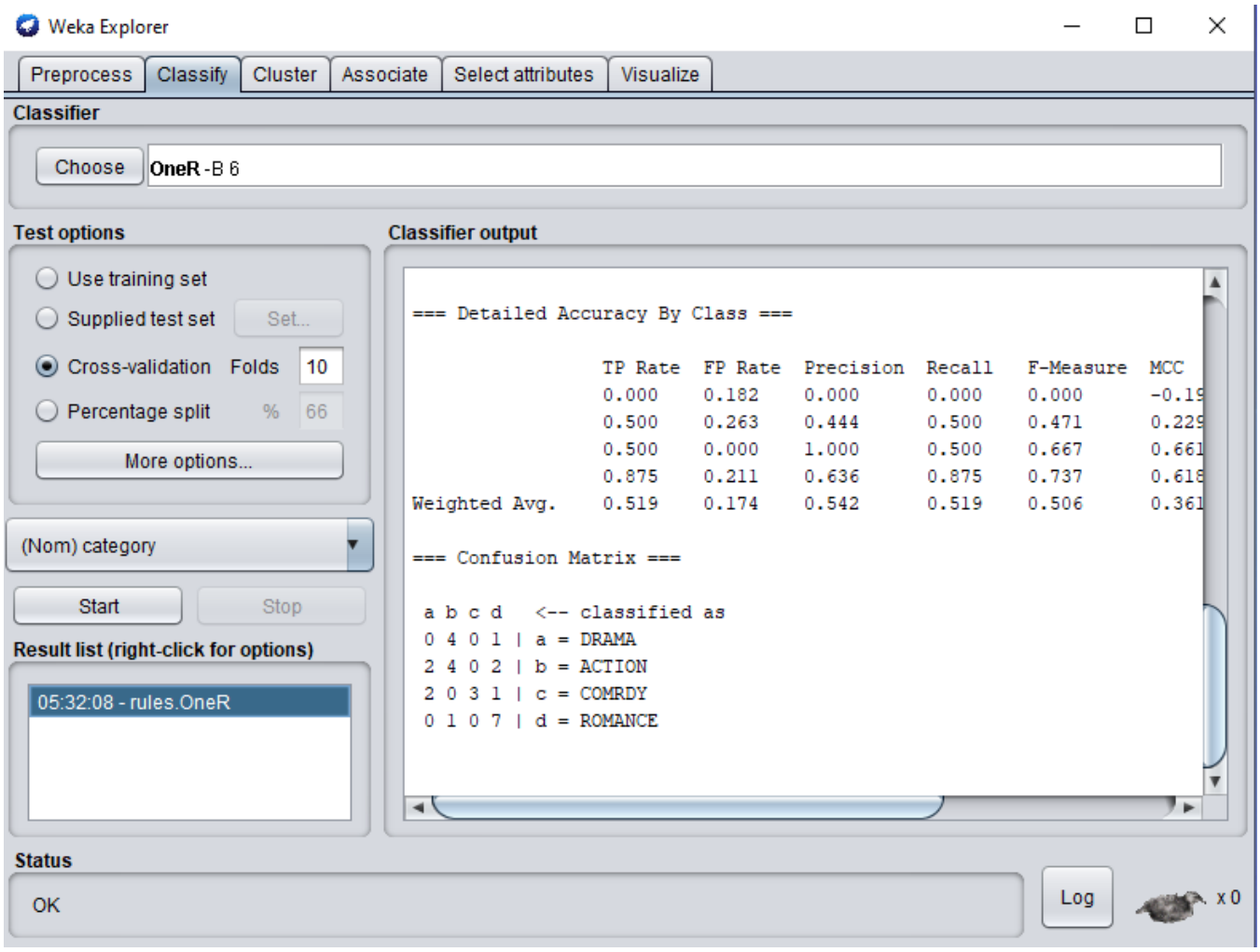


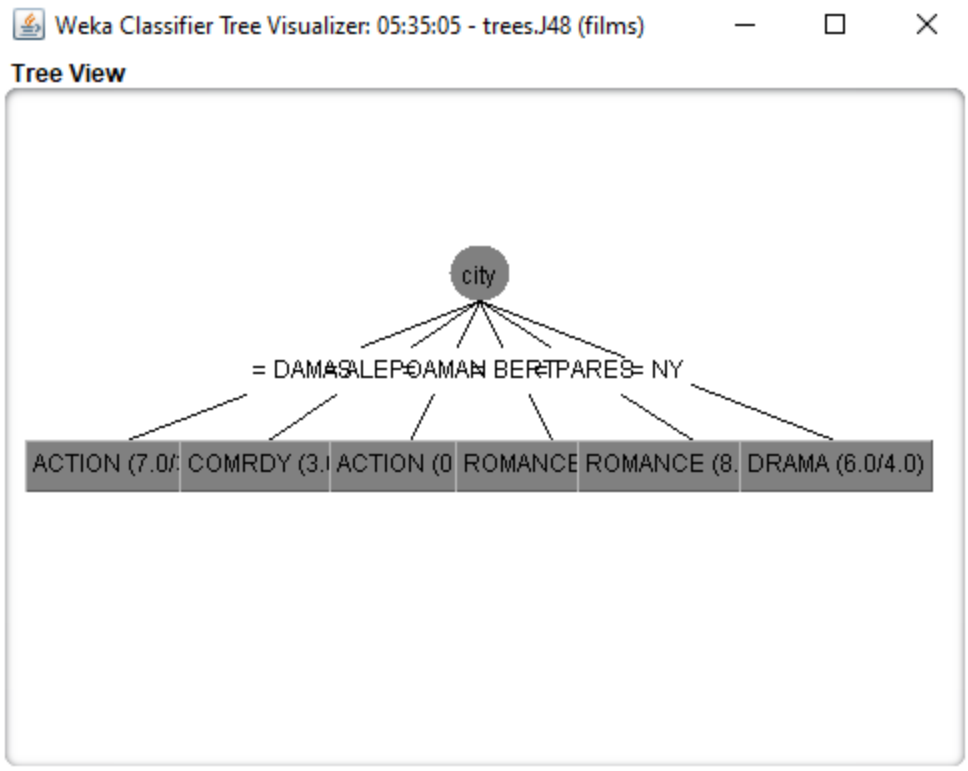

```
Materialized view CUST_FILM_DATA created.
```

- الطلب السادس :

```
@RELATION films

@attribute 'city' {DAMAS,ALEPO,AMAN,BERT,PARES,NY}
@attribute 'lanuage' {AR,EN,FR}
@attribute 'category' {DRAMA,ACTION,COMRDY,ROMANCE}

@DATA
DAMAS,AR,DRAMA
DAMAS,AR,ACTION
ALEPO,AR,COMRDY
BERT,EN,ROMANCE
NY,EN,COMRDY
NY,EN,COMRDY
PARES,FR,ROMANCE
PARES,FR,COMRDY
DAMAS,AR,ACTION
DAMAS,AR,ACTION
ALEPO,AR,COMRDY
BERT,EN,ROMANCE
NY,EN,DRAMA
NY,EN,DRAMA
PARES,FR,ROMANCE
PARES,AR,DRAMA
DAMAS,AR,DRAMA
PARES,FR,ACTION
PARES,EN,ROMANCE
DAMAS,AR,ROMANCE
DAMAS,AR,ACTION
ALEPO,AR,COMRDY
BERT,EN,ROMANCE
NY,EN,ACTION
NY,EN,ACTION
PARES,FR,ROMANCE
PARES,FR,ACTION
```


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: films
Instances: 27
Attributes: 3
Sum of weights: 27
Attributes
All
None
Invert
Pattern
No.
Name
1 city
2 lanuage
3 category
Remove
Selected attribute
Name: city
Missing: 0 (0%)
Distinct: 5
Type: Nominal
Unique: 0 (0%)
No.
Label
Count
Weight
1 DAMAS 7 7.0
2 ALEPO 3 3.0
3 AMAN 0 0.0
4 BERT 3 3.0
Class: category (Nom)
Visualize All
7
3
0
3
8
6
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
OneR -B 6
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.000 0.182 0.000 0.000 0.000 -0.19
0.500 0.263 0.444 0.500 0.471 0.229
0.500 0.000 1.000 0.500 0.667 0.661
0.875 0.211 0.636 0.875 0.737 0.618
Weighted Avg. 0.519 0.174 0.542 0.519 0.506 0.361
=== Confusion Matrix ===
a b c d <-- classified as
0 4 0 1 | a = DRAMA
2 4 0 2 | b = ACTION
2 0 3 1 | c = COMRDY
0 1 0 7 | d = ROMANCE
(Nom) category
Start
Stop
Result list (right-click for options)
05:32:08 - rules.OneR
Status
OK
Log
x 0


Weka Classifier Tree Visualizer: 05:35:05 - trees.J48 (films)
Tree View
city
= DAMASALEPOAMAN BERITPARES= NY
ACTION (7.0/
COMRDY (3.
ACTION (0
ROMANCE
ROMANCE (8.
DRAMA (6.0/4.0)