**DATA WAREHOUSE DESIGN AND OLAP**

**IMAT5167\_2223\_501**

**P2736418 -Aathira Puthenpurayil**

**Data Mart design Assignment**

**9th December 2022**

**Lecturer: Professor Yingjie Yang**

**Midlands Theatre (MT) Company**

**Task 1:**

Analyse the given database design and the requirement for a data mart and identify the dimensions and fact for your data mart. Here for this assignment, only the essential (minimum) dimensions and attributes for the required queries need to be included.

1.Dimensions identified for the data mart:

* P2736418\_dp is the dimension for Production
* P2736418\_dth is the dimension for Theatre
* P2736418\_dc is the dimension for client
* P2736418\_dt is the dimension for time

2.Fact identified for the data mart:

* P2736418\_Ftp is the dimension for the fact

**Justification for data mart selected: -** we need to analyse production attributes from production table, theatre attributes from theatre table and highest spend client from client table.

Here monthly sale can be obtained by extracting month & year from pDate in performance table to separate dimension for time table.

Highest spend for each client can be obtained from the TotalAmount in ticket purchase table extracted to fact table.

**Task 2:**

Design the star schema for the data mart and determine the relevant attributes and suitable granularity.

**Relevant attributes**

* For star schema p#,theatre#,client# are the natural keys for the dimensions
* production\_id,theatre\_id,client\_id and time\_id are the surrogate keys for the dimensions.
* For P2736418\_dp, the PK is production\_id
* For P2736418\_dth, the PK is theatre\_id
* For P2736418\_dc, the PK is client\_id
* For P2736418\_dt, the PK is time\_id
* For Ftp the PK is a composite key consisting of theatre\_id, production\_id,client\_id and time\_id.
* Theatre\_id is a FK referring to P2736418\_dth
* Production\_id\_is a FK referring to P2736418\_dp
* Client\_id is a FK referring to P2736418\_dc
* Time\_id is a FK referring to P2736418\_dt

**Granularity**

The granularity for each row in ftp represents the TotalAmount spend by each client in each theatre in each month.

**Star Schema**

P2736418\_ftp

production\_id (FK)

client\_id (FK)

theatre\_id (FK)

time\_id (FK)

TotalAmount

P2736418\_dp

production\_id (PK)

p#

title

P2736418\_dth

theatre\_id (PK)

theatre#

name

P2736418\_dt

time\_id (PK)

year

month

P2736418\_dc

client\_id (PK)

client#

title

name

**Task 3:**

Map your star schema to logical relations and identify the corresponding PKs and FKs.

**Logical relations**

P2736418\_dp(production\_id,p#,title)

P2736418\_dth(theatre\_id,theatre#,name)

P2736418\_dc(client\_id,client#,title,name)

P2736418\_dt(time\_id,year,month)

P2736418\_ftp(production\_id,theatre\_id,client\_id,time\_id,TotalAmount)

* production\_id is a FK to P2736418\_dp
* theatre\_id is a FK to P2736418\_dth
* client\_id is a FK to P2736418\_dc
* time\_id is a FK to P2736418\_dt

**Task 4:**

* Create the corresponding tables in Oracle using SQL

**Theatre**

1. **Creating sequence for theatre**

CREATE SEQUENCE P2736418\_dth\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

Graphical user interface, text, application

Description automatically generated

**2**.**Create the theatre table in Oracle using SQL.**

CREATE TABLE P2736418\_dth (

theatre\_id number(5) PRIMARY KEY,

theatre# VARCHAR2(5) NOT NULL,

name VARCHAR2(20) NOT NULL);

Text

Description automatically generated

**Production**

1. **Create sequence**

CREATE SEQUENCE P2736418\_dp\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

Graphical user interface, text, application

Description automatically generated

**2.create Table**

CREATE TABLE P2736418\_dp (

production\_id number(5) PRIMARY KEY,

p# VARCHAR2(5) NOT NULL,

title VARCHAR2(20) NOT NULL);

Text

Description automatically generated

**Client**

**1.create sequence**

CREATE SEQUENCE P2736418\_dc\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

Graphical user interface, text, application

Description automatically generated

**2.create Table**

CREATE TABLE P2736418\_dc (

client\_id number(5) PRIMARY KEY,

client# VARCHAR2(5) NOT NULL,

title VARCHAR2(10) NOT NULL,

name VARCHAR2(30) NOT NULL);

Graphical user interface, text, application

Description automatically generated

**Time Table**

**1.Create sequence**

CREATE SEQUENCE P2736418\_dt\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

Graphical user interface, text, application

Description automatically generated

**2.Create table**

CREATE TABLE P2736418\_dt (

time\_id number (5) PRIMARY KEY,

year number (4) NOT NULL,

month number (2) NOT NULL);

Graphical user interface, text

Description automatically generated

**Fact table**

**1.create table**

CREATE TABLE P2736418\_ftp(

production\_id number(5) CONSTRAINT fk3 REFERENCES P2736418\_dp,

theatre\_id number(5) CONSTRAINT fk4 REFERENCES P2736418\_dth,

client\_id number(5) CONSTRAINT fk5 REFERENCES P2736418\_dc,

time\_id number(5) CONSTRAINT fk7 REFERENCES P2736418\_dt,

TotalAmount NUMBER(6,2) NOT NULL,

constraint pk2\_fsp primary key (production\_id,theatre\_id,client\_id,time\_id));

Text

Description automatically generated

**Task 5:**

* Identify your source data from the OLTP database and design your data extraction rules. You need to give a detailed mapping and transformation list from the source to the destination.

**Theatre Mapping & transformation**

Diagram

Description automatically generated

Source contains Theatre,performance and ticket purchase table which has been joined and destination has dimension table for theatre. Need to format the name ,extract month from pDate and sum the total amount from ftp.

**Production Mapping & transformation**

Diagram

Description automatically generated

Source contains production, performance and ticket purchase table which has been joined and destination has dimension table for production. Need to format the title, extract month from pDate and sum the total amount from ftp.

**Client Mapping & transformation**

Diagram

Description automatically generated

Source contains client and ticket purchase table which has been joined and destination has dimension table for client. Need to format the name ,extract month from pDate and sum the total amount from ftp.

**Task 6:**

Implement your data extraction, transformation and loading through Oracle SQL. The number of rows extracted into each dimension or fact table in your data mart should be printed from Oracle query.

**Theatre table**

**Insert the values into theatre table**

insert into P2736418\_dth select P2736418\_dth\_seq.nextval, theatre#, name from

(select distinct theatre#, upper(trim(name)) name from

(select th.theatre#, th.name

From ops$yyang00.theatre th, ops$yyang00.performance per,ops$yyang00.ticketpurchase tp

Where th.theatre#=per.theatre# and per.per#=tp.per#));

Graphical user interface, text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

For theatre table 6 rows has been returned. (select \* from P2736418\_dth;)

**Production Table**

**Insert values**

insert into P2736418\_dp select P2736418\_dp\_seq.nextval, p#, title from

(select distinct p#, upper(trim(title)) title from

(select p.p#, p.title

From ops$yyang00.production p, ops$yyang00.performance per,ops$yyang00.ticketpurchase tp

Where p.p#=per.p# and per.per#=tp.per#));

Graphical user interface, text

Description automatically generated

Table

Description automatically generated with medium confidence

For production table 78 rows has been returned. (select \* from P2736418\_dp;)

**Client Table**

**insert Values**

insert into P2736418\_dc select P2736418\_dc\_seq.nextval, client#, title, name from

(select distinct client#, upper(trim(title)) title, upper(trim(name)) name from

(select c.client#, c.title ,c.name

From ops$yyang00.client c,ops$yyang00.ticketpurchase tp

Where c.client#=tp.client#));

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

For the client table 3117 rows has been returned. (select \* from P2736418\_dc;)

**Time table**

**Insert value**

insert into P2736418\_dt select P2736418\_dt\_seq.nextval, year, month from

(select distinct extract(year from pDate) year, extract(month from pDate) month from ops$yyang00.performance);

Graphical user interface

Description automatically generated with medium confidence

A picture containing graphical user interface

Description automatically generated

For the time table 17 rows has been returned. (select \* from P2736418\_dt;)

**Fact Table**

**Insert values**

insert into P2736418\_ftp select production\_id, theatre\_id, client\_id, time\_id,total from

(select P2736418\_dp.production\_id, P2736418\_dth.theatre\_id, P2736418\_dc.client\_id, P2736418\_dt.time\_id, sum(TotalAmount) total

from ops$yyang00.performance per, ops$yyang00.ticketpurchase tp, P2736418\_dp, P2736418\_dth,P2736418\_dc,P2736418\_dt

Where per.per#=tp.per# and P2736418\_dth.theatre#=per.theatre# and per.p#=P2736418\_dp.p# and tp.client#=P2736418\_dc.client# and

extract(year from per.pDate)=P2736418\_dt.year and extract(month from per.pDate)=P2736418\_dt.month

Group by production\_id, theatre\_id, client\_id, time\_id);

Graphical user interface, application, Word

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

For the fact table 4811 rows has been returned.

**Task 7:**

Comment on how your data mart satisfies the requirements of MT. Implement the required data analysis requirements for both the data mart and the original OLTP database. Compare your queries and comment on the advantages of a data mart in analysis operations. Test results from Oracle query should be included.

**Question**

**1.The total sale value of each production.**

**Data Mart**

Select P2736418\_dp.production\_id, P2736418\_dp.title, sum(TotalAmount) as total\_sale

from P2736418\_dp,P2736418\_dt, P2736418\_ftp

where P2736418\_dp.production\_id=P2736418\_ftp.production\_id and P2736418\_dt.time\_id=P2736418\_ftp.time\_id

group by P2736418\_dp.production\_id, P2736418\_dp.title;

Table

Description automatically generated

**Relational database**

select p.p#, p.title, sum(TotalAmount) as total\_sale

from ops$yyang00.production p, ops$yyang00.performance per, ops$yyang00.ticketpurchase tp

where p.p#=per.p# and per.per#=tp.per#

group by p.p#, p.title;

Graphical user interface, text, application, email

Description automatically generated

**Comparison of Data Mart and Relational Model**

In Data mart we can create with only required rows for our query. Query is more readable ad easy to script compared to relational model.

**2.Monthly sale value of each theatre.**

**Data Mart**

select P2736418\_dth.theatre\_id, P2736418\_dth.name, P2736418\_dt.year, P2736418\_dt.month, sum(TotalAmount) as Monthly\_sale from P2736418\_dth,P2736418\_dt, P2736418\_ftp where P2736418\_dth.theatre\_id=P2736418\_ftp.theatre\_id and P2736418\_dt.time\_id=P2736418\_ftp.time\_id group by P2736418\_dth.theatre\_id, P2736418\_dth.name, P2736418\_dt.year, P2736418\_dt.month ;

Table

Description automatically generated

**Relational database**

select th.theatre#, th.name, extract(year from pDate) as year, extract(month from pDate) as month, sum(TotalAmount) as Monthly\_sale

from ops$yyang00.theatre th, ops$yyang00.performance per, ops$yyang00.ticketpurchase tp

where th.theatre#=per.theatre# and per.per#=tp.per#

group by th.theatre#, th.name, extract(year from pDate), extract(month from pDate);

A screenshot of a computer

Description automatically generated with medium confidence

**Comparison of Data Mart and relational model**

In relational data base we need to access ticket purchase, performance to get monthly sale of each theatre but in data mart it is more simplified as the required column for the requirement is derived from fact table that makes join simpler.

**3.The theatre name (for each theatre) and the names of clients who have the highest spending in that theatre.**

**Data mart**

select theatre\_name, client\_name,total\_amount

from (select sum(f.TotalAmount) as total\_amount, max(th.name) as theatre\_name,max(c.name) as client\_name, dense\_rank() over (partition by th.theatre\_id order by sum(f.TotalAmount)desc ) as rank

from P2736418\_ftp f

inner join P2736418\_dth th on (f.theatre\_id = th.theatre\_id)

inner join P2736418\_dc c on (c.client\_id = f.client\_id)

group by th.theatre\_id , c.client\_id)

where rank = 1;

Graphical user interface, table

Description automatically generated with medium confidence

**Relational model**

select theatre\_name, client\_name, total\_amount

from (select sum (tp.TotalAmount) as total\_amount , max(th.name) as theatre\_name, max(c.name) as client\_name,dense\_rank() over (partition by th.theatre# order by sum(tp.TotalAmount)desc) as rank

from ops$yyang00.performance per

inner join ops$yyang00.theatre th on (th.theatre# = per.theatre#)

inner join ops$yyang00.ticketpurchase tp on (per.per# = tp.per#)

inner join ops$yyang00.client c on (c.client# = tp.client#)

group by th.theatre#,c.client#)

where rank = 1;

Graphical user interface, table

Description automatically generated

**Comparison of Data Mart and relational model**

Execution time is same for both data mart and relational model. But in data mart it is easy to execute the inner join.