**MIDLANDS THEATRE(MT) COMPANY**

Data Mart Design

**ASSIGNMENT BY ANNE AWELE NWAOKOLO- P2743914**

**Data Warehouse Design and OLAP**

**IMAT 5167**

**DELIVERABLES**

**1.1 STAR SCHEMA DESIGN**

**P2743914­\_CLIENT**

**Dimension**

**Client\_id PK**

Client#

**Title**

Name

**Theatre\_Id PK**

Theatre#

Name

**P2743914\_THEATRE**

Dimension

**1.2 EXPLANATION OF STAR SCHEMA**

**P2743914\_TIME**

**Dimension**

**P2743914­­\_TICKETSALES**

**Fact**

Theatre\_Id FK

Prod\_Id FK

Purchase**\_id**, FK

time\_id FK

SalesValue

**P2743914\_PRODUCTION**

**Dimension**

**Prod\_Id PK**

P#

Title

**Time\_id PK**

Month

year

**a and b**

**P2743914\_theatre – Dimension-** This table is required to get the details of the theatre and the required attributes are **Theatre\_Id, Theatre#, name**. The table and the attributes are chosen because the analysis requires us getting the details of theatre with the highest spending.

**P2743914\_production – Dimension**- This table is required to enable us get information on sales for each production as required in the analysis.

Required attributes are **P\_Id, P#, Title** all these contain information on the production and will be used in the analysis.

**P2743914\_Client – Dimension-** This table is required to extract information for our clients as the analysis requires finding out which clients have the highest spending. attributes required are **Client\_id , PK Client# ,Title, Name** each of these attributes give us information of clients as expected in the analysis.

**P2743914\_time – Dimension-** in the analysis, we are expected to get monthly sales value of each theatre so a table with time details is needed for this analysis and the required attrbutes are **Time\_id, month, year**

**P2743914\_Ticketsales-Fact-** I have chosen ticketsales as my fact table this is because the main requirement of all the analysis borders around getting the value of sales for tickets and I have choses to include all the primary keys from my dimension tables to complete the star schema, these foreign keys are referencing the primary keys from all the dimension tables the attrtibutes include **Theatre\_Id, P\_Id,Purchase\_id, time\_id , SalesValue** I included an additional attribute for the value of sales as the fact data.

**c. Granularity**

Time grain- Month

Theatre grain-Total sales value specific theatre

production grain -Monthly sales value for a specific production

theatre client-specific theatre and specific clients with highest spending

**2. DERIVATION OF LOGICAL RELATIONS**

**2.1 LIST OF LOGICAL RELATIONS**

**P2743914\_theatre –** (Theatre\_Id PK, Theatre#, name)

**P2743914\_production –** (P\_Id PK **,** P#**,** Title)

**P2743914\_Client –** (Client\_id PK, Client# ,Title, Name )

**P2743914\_time – Dimension-** (time\_id PK, month, year)

**P2743914\_Ticketsales-Fact-** (Theatre\_Id FK, Prod\_Id FK ,Purchase\_id FK, time\_id FK , SalesValue)

**2.2 EXPLANATION ON THE MAPPING FROM STAR SCHEMA TO THE LOGICAL RELATIONS**

Theatre\_Id is a foreign key referring to the dimension table P2743914\_theatre

P\_Id is a foreign key referring to the dimension table P2743914\_production

Client\_id is a foreign key referring to the dimension table P2743914\_ Client

time\_id is a foreign key referring to the dimension table P2743914\_time

these foreign keys form the composition of the primary keys of the fact table P2743914\_Ticketsales.

Theatre\_Id, P\_Id, Client\_id, time\_id are surrogate keys while Theatre#, P#, Client# are the Nature keys

**3. CREATION OF TABLES**

**P2743914\_THEATRE TABLE**

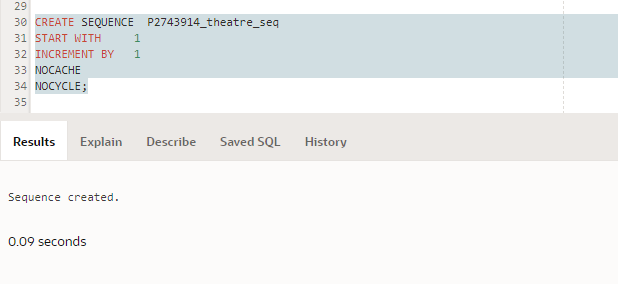
CREATE SEQUENCE P2743914\_theatre\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

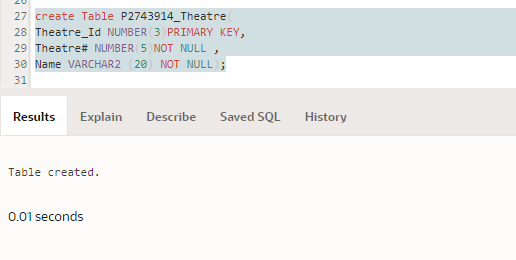
****

create Table P2743914\_Theatre(

Theatre\_Id NUMBER(3)PRIMARY KEY,

Theatre# NUMBER(5)NOT NULL ,

Name VARCHAR2 (20) NOT NULL);

****

**P2743914\_PRODUCTION TABLE**

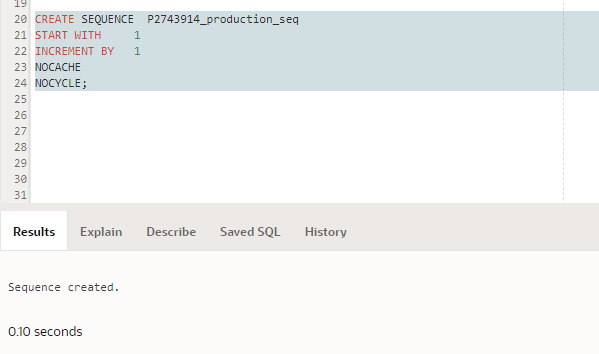
CREATE SEQUENCE P2743914\_production\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

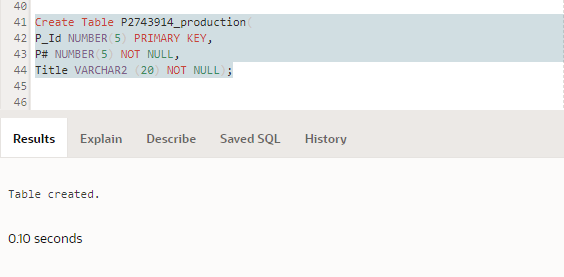
NOCYCLE;

****

Create Table P2743914\_production(

P\_Id NUMBER(5) PRIMARY KEY,

P# NUMBER(5) NOT NULL,

Title VARCHAR2 (20) NOT NULL);****

**P27439214\_CLIENT TABLE**

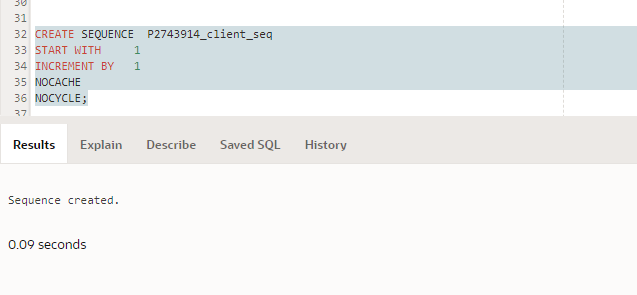
CREATE SEQUENCE P2743914\_client\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

****

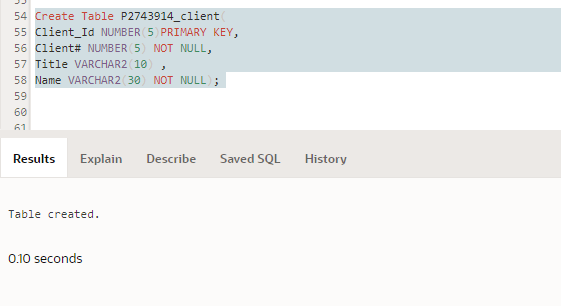
Create Table P2743914\_client(

Client\_Id NUMBER(5)PRIMARY KEY,

Client# NUMBER(5) NOT NULL,

Title VARCHAR2(10),

Name VARCHAR2(30) NOT NULL);

****

**P2743914\_TIME TABLE**

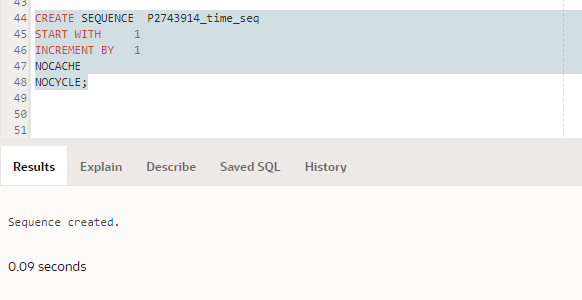
CREATE SEQUENCE P2743914\_time\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

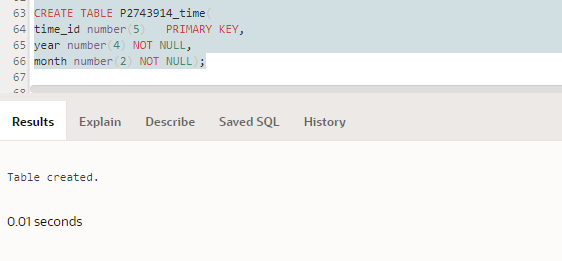
****

CREATE TABLE P2743914\_time(

time\_id number(5)PRIMARY KEY,

year number(4) NOT NULL,

month number(2) NOT NULL);

****

**P2743914\_TICKETSALES TABLE**

create Table P2743914\_ticketsales(

THEATRE\_Id NUMBER(5) CONSTRAINT FK\_P2\_ticketsales REFERENCES P2743914\_Theatre,

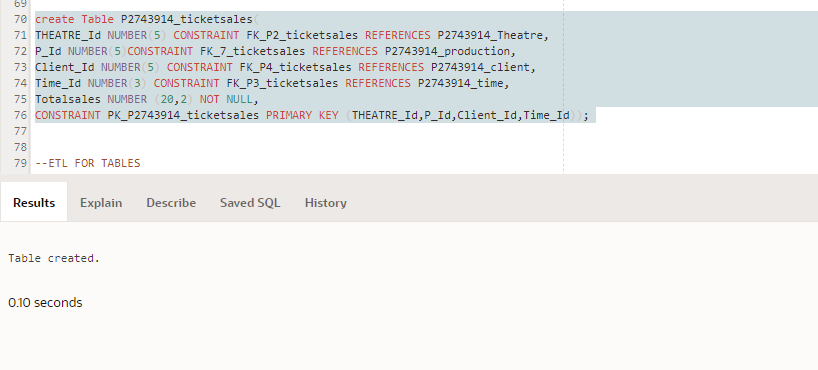
P\_Id NUMBER(5)CONSTRAINT FK\_7\_ticketsales REFERENCES P2743914\_production,

Client\_Id NUMBER(5) CONSTRAINT FK\_P4\_ticketsales REFERENCES P2743914\_client,

Time\_Id NUMBER(3) CONSTRAINT FK\_P3\_ticketsales REFERENCES P2743914\_time,

Totalsales NUMBER (20,2) NOT NULL,

CONSTRAINT PK\_P2743914\_ticketsales PRIMARY KEY (THEATRE\_Id,P\_Id,Client\_Id,Time\_Id));

****

**4. DATA SOURCE IDENTIFICATION, DATA EXTRACTION, TRANSFORMATION AND LOADING**

**4.1 DATA SOURCE MAPPING**

Theatre joins Performance

joins TicketPurchase

Theatre#

name

…

pdate

Totalamount

P2743914\_Theatre

Theatre#

name

Format

Format

P2743914\_time

time\_id

year

month

P2743914\_Ticketsales

Time\_id

Totalsales

Production joins performance joins Ticket Purchase

P#

Title

…

pdate

Totalamount

P2743914\_Production

P#

title

Format

Format

P2743914\_time

time\_id

year

month

P2743914\_Ticketsales

Time\_id

Totalsales

Client join ticketpurchase

Client#

Title

name

…

Totalamount

P2743914\_client

Client#

Title

name

Format

Format

P2743914\_Ticketsales

Totalsales

**4.2 AND 4.3 – ETL CODES AND EVIDENCE OF CODES WORKING(SCREENSHOT)**

**P2743914\_THEATRE**

insert into P2743914\_Theatre select P2743914\_Theatre\_seq.nextval, Theatre#, name from

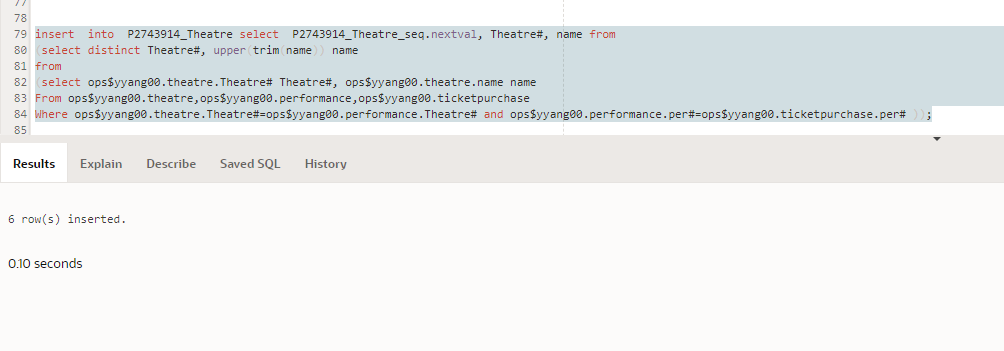
(select distinct Theatre#, upper(trim(name)) name

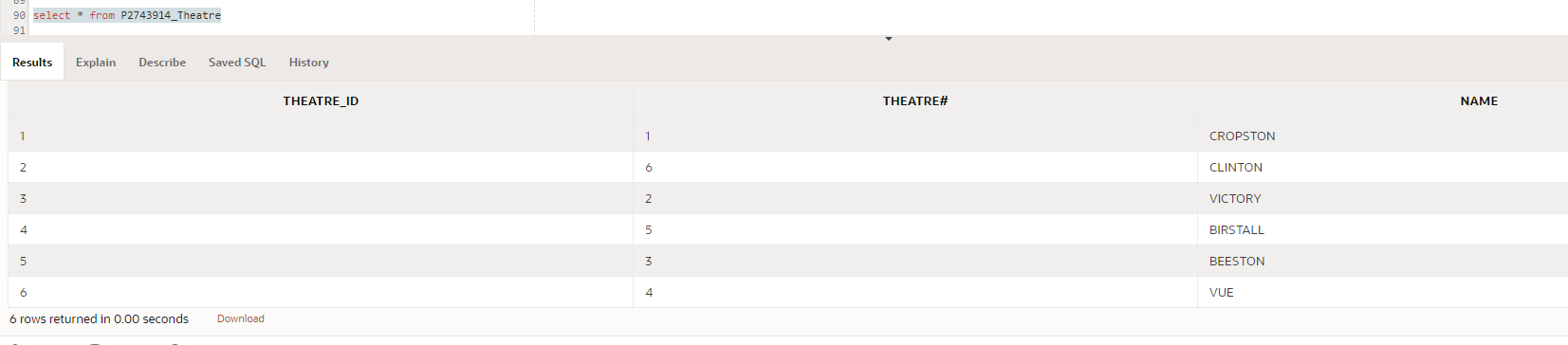
from

(select ops$yyang00.theatre.Theatre# Theatre#, ops$yyang00.theatre.name name

From ops$yyang00.theatre,ops$yyang00.performance,ops$yyang00.ticketpurchase

Where ops$yyang00.theatre.Theatre#=ops$yyang00.performance.Theatre# and ops$yyang00.performance.per#=ops$yyang00.ticketpurchase.per# ));

****

****

**P2743914\_PRODUCTION**

insert into P2743914\_Production select P2743914\_Production\_seq.nextval, P#, title from

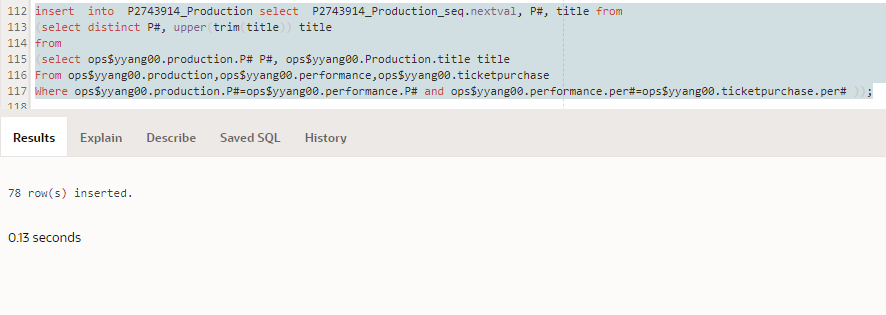
(select distinct P#, upper(trim(title)) title

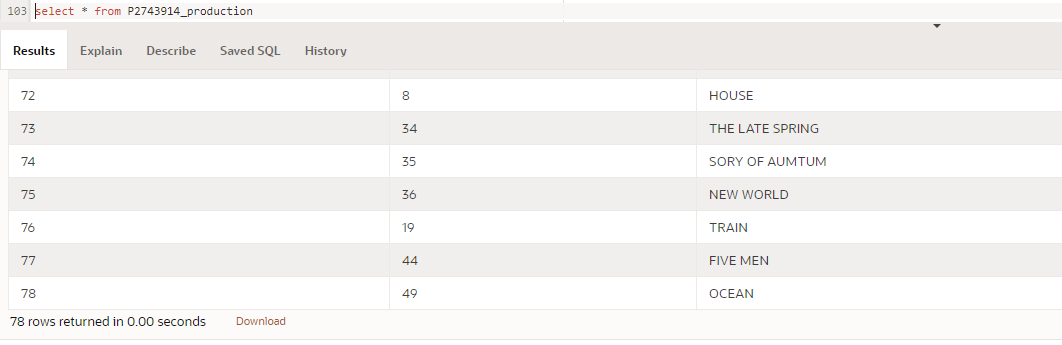
from

(select ops$yyang00.production.P# P#, ops$yyang00.Production.title title

From ops$yyang00.production,ops$yyang00.performance,ops$yyang00.ticketpurchase

Where ops$yyang00.production.P#=ops$yyang00.performance.P# and ops$yyang00.performance.per#=ops$yyang00.ticketpurchase.per# ));

****

****

**P2743914\_CLIENT**

insert into P2743914\_Client select P2743914\_client\_seq.nextval, Client#,Title,Name from

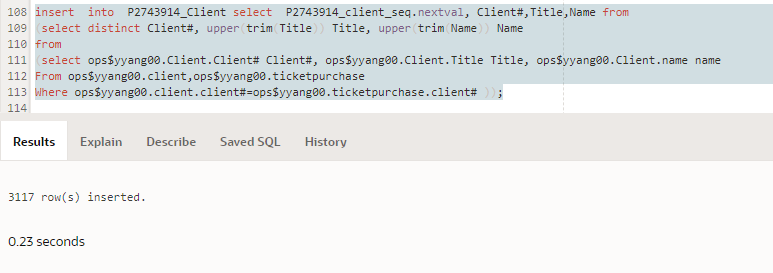
(select distinct Client#, upper(trim(Title)) Title, upper(trim(Name)) Name

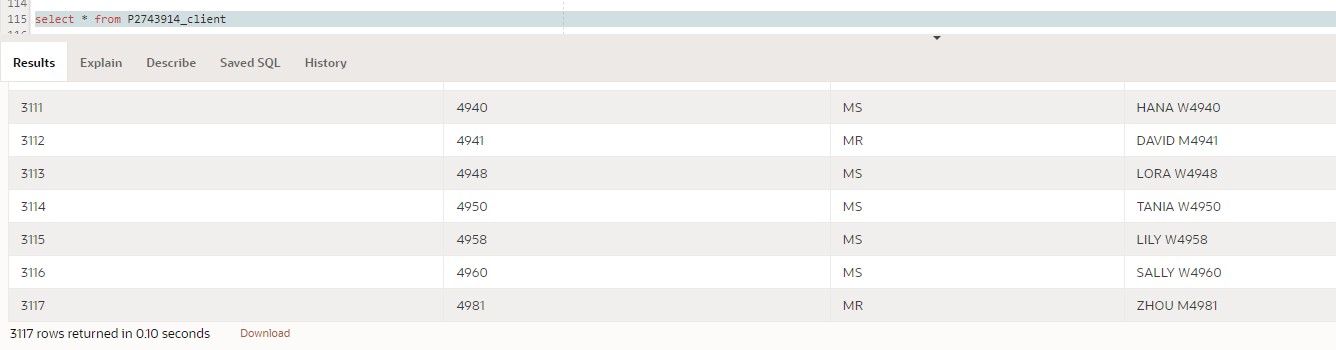
from

(select ops$yyang00.Client.Client# Client#, ops$yyang00.Client.Title Title, ops$yyang00.Client.name name

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

Where ops$yyang00.client.client#=ops$yyang00.ticketpurchase.client# ));

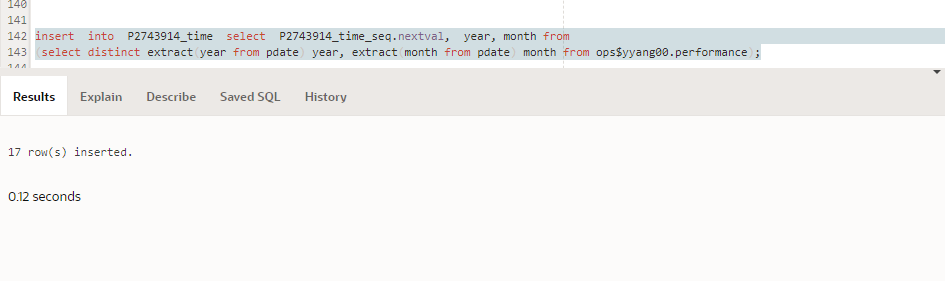
****

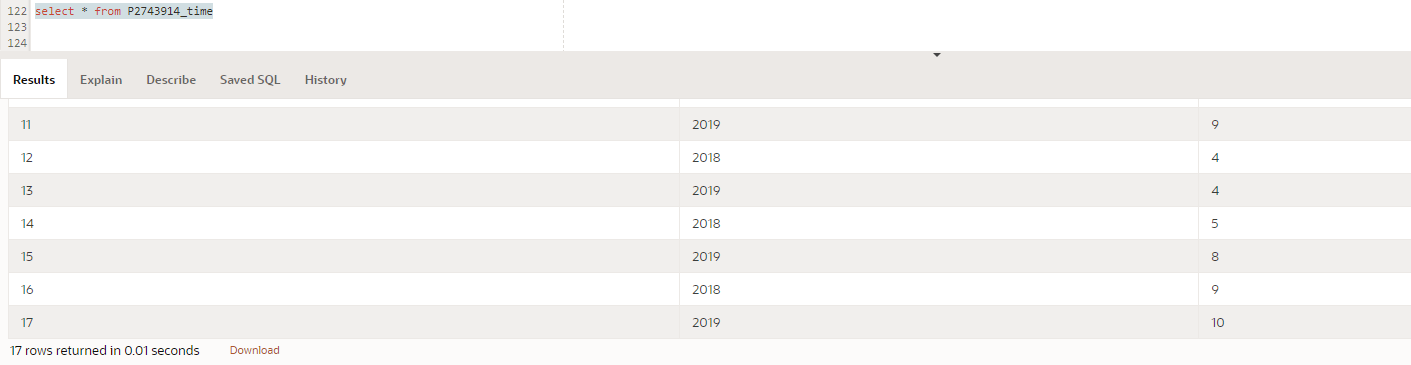
****

**P2743914\_TIME**

insert into P2743914\_time select P2743914\_time\_seq.nextval, year, month from

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

****

****

**P2743914\_TICKETSALES**

Insert into P2743914\_ticketsales Select Theatre\_Id,P\_Id,Client\_Id,Time\_Id, totalsales from

(select P2743914\_Client.Client\_id, P2743914\_Theatre.Theatre\_id,P2743914\_Production.P\_Id, P2743914\_time.Time\_Id, sum(totalamount)Totalsales

from P2743914\_Client, P2743914\_Theatre,P2743914\_time,P2743914\_Production,ops$yyang00.ticketpurchase,ops$yyang00.performance

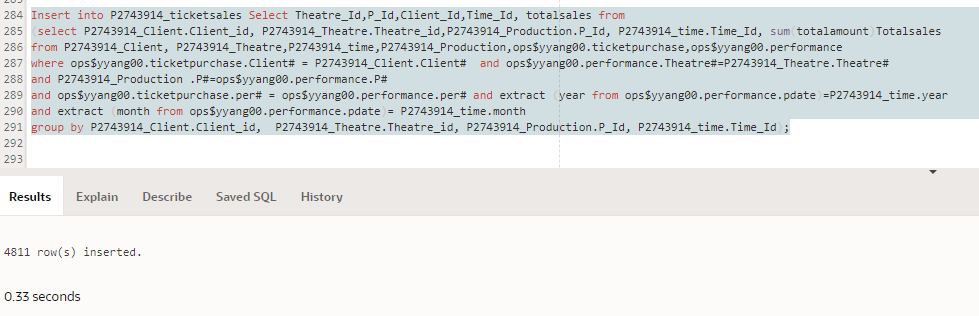
where ops$yyang00.ticketpurchase.Client# = P2743914\_Client.Client# and ops$yyang00.performance.Theatre#=P2743914\_Theatre.Theatre#

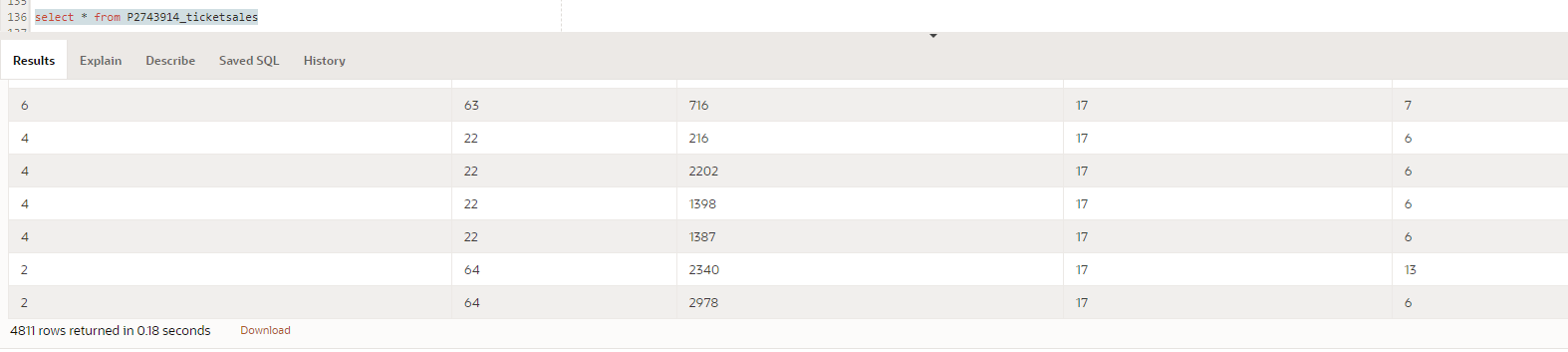
and P2743914\_Production .P#=ops$yyang00.performance.P#

and ops$yyang00.ticketpurchase.per# = ops$yyang00.performance.per# and extract (year from ops$yyang00.performance.pdate)=P2743914\_time.year

and extract (month from ops$yyang00.performance.pdate)= P2743914\_time.month

group by P2743914\_Client.Client\_id, P2743914\_Theatre.Theatre\_id, P2743914\_Production.P\_Id, P2743914\_time.Time\_Id);

****

****

**5. JUSTIFICATION OF THE DATA MART DESIGN AND COMPARISON OF DATA MART AND OLTP**

**5.1 SQL CODE FOR THE REQUIRED QUERIES**

**ANALYSING TOTAL SALES VALUE OF EACH PRODUCTION**

**DATA MART**

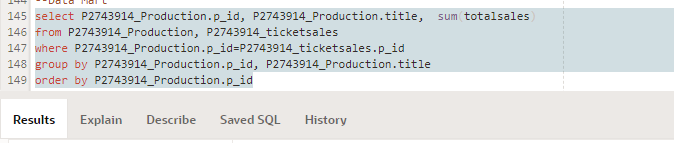
select P2743914\_Production.p\_id, P2743914\_Production.title, sum(totalsales)

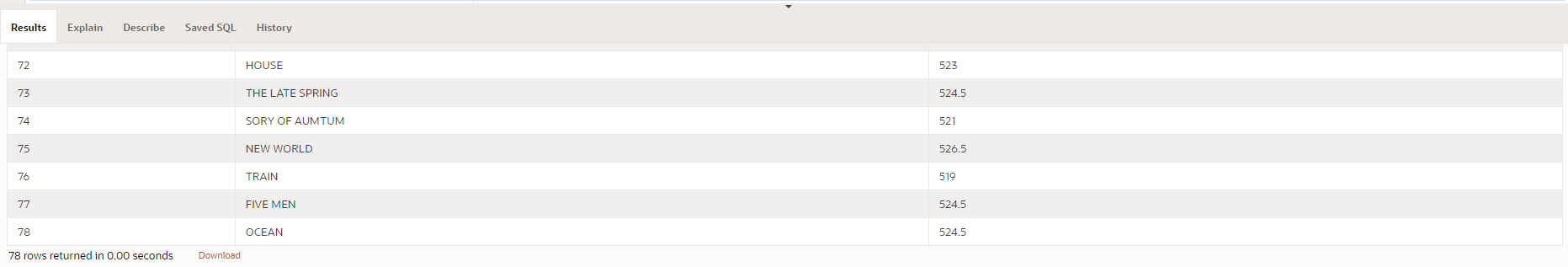
from P2743914\_Production, P2743914\_ticketsales

where P2743914\_Production.p\_id=P2743914\_ticketsales.p\_id

group by P2743914\_Production.p\_id, P2743914\_Production.title

order by P2743914\_Production.p\_id

****

****

**RELATIONAL MODEL**

select ops$yyang00.production.P# P#, ops$yyang00.Production.title title,sum(totalamount)

From ops$yyang00.production,ops$yyang00.performance,ops$yyang00.ticketpurchase

Where ops$yyang00.production.P#=ops$yyang00.performance.P# and ops$yyang00.performance.per#=ops$yyang00.ticketpurchase.per#

group by ops$yyang00.production.p#, ops$yyang00.production.title

order by ops$yyang00.production.p#

****

**ANALYSING MONTHLY SALE VALUE OF EACH THEATRE**

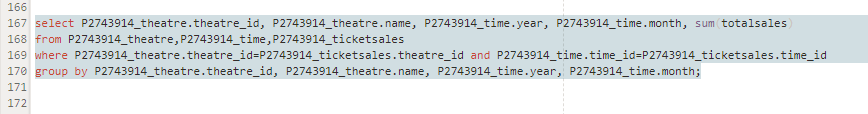
**DATA MART**

select P2743914\_theatre.theatre\_id, P2743914\_theatre.name, P2743914\_time.year, P2743914\_time.month, sum(totalsales)

from P2743914\_theatre,P2743914\_time,P2743914\_ticketsales

where P2743914\_theatre.theatre\_id=P2743914\_ticketsales.theatre\_id and P2743914\_time.time\_id=P2743914\_ticketsales.time\_id

group by P2743914\_theatre.theatre\_id, P2743914\_theatre.name, P2743914\_time.year, P2743914\_time.month;





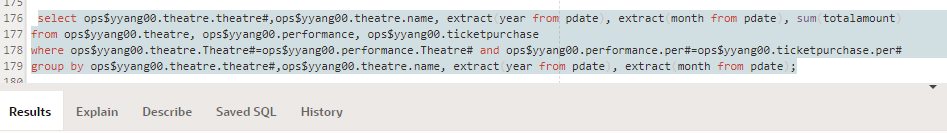
**RELATIONAL DATA MODEL**

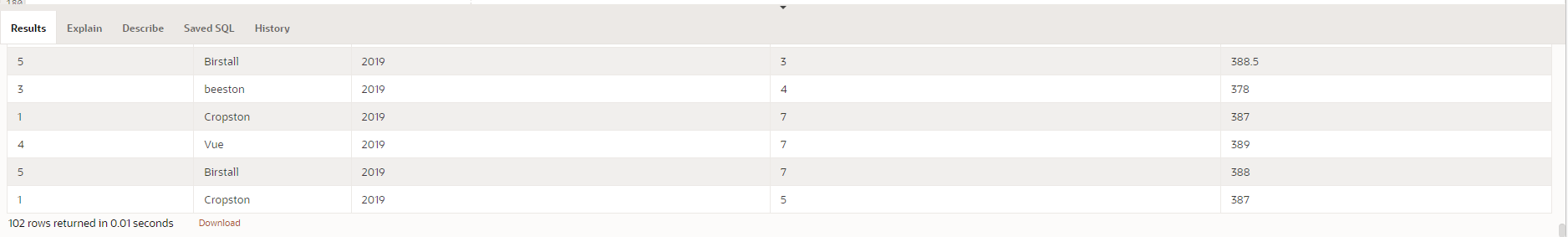
select ops$yyang00.theatre.theatre#,ops$yyang00.theatre.name, extract(year from pdate), extract(month from pdate), sum(totalamount)

from ops$yyang00.theatre, ops$yyang00.performance, ops$yyang00.ticketpurchase

where ops$yyang00.theatre.Theatre#=ops$yyang00.performance.Theatre# and ops$yyang00.performance.per#=ops$yyang00.ticketpurchase.per#

group by ops$yyang00.theatre.theatre#,ops$yyang00.theatre.name, extract(year from pdate), extract(month from pdate);

****

****

**ANALYSING THE CLIENT NAME WITH HIGHEST SPENDING IN EACH THEATRE**

select T1.name,T1.theatre\_id,TS1.totalsales,C1.name,C1.client\_id from P2743914\_client C1, P2743914\_theatre T1, P2743914\_ticketsales TS1 where totalsales=(

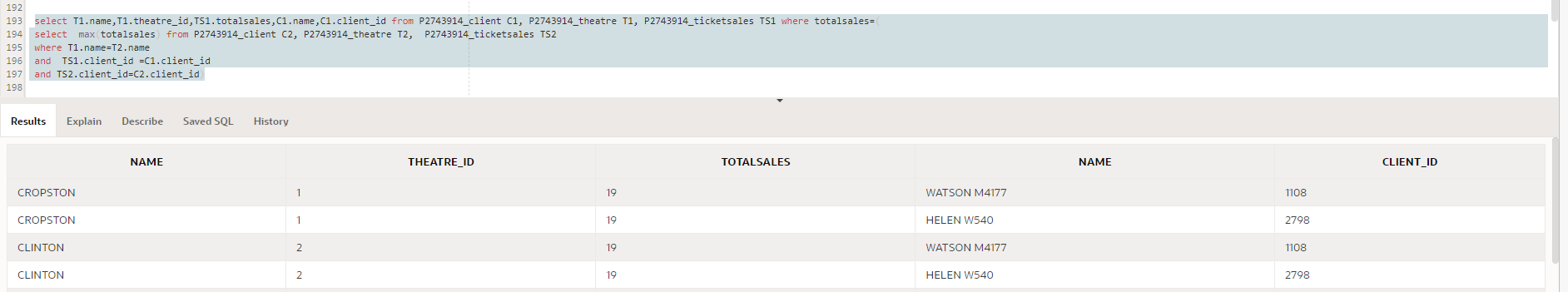
select max(totalsales) from P2743914\_client C2, P2743914\_theatre T2, P2743914\_ticketsales TS2

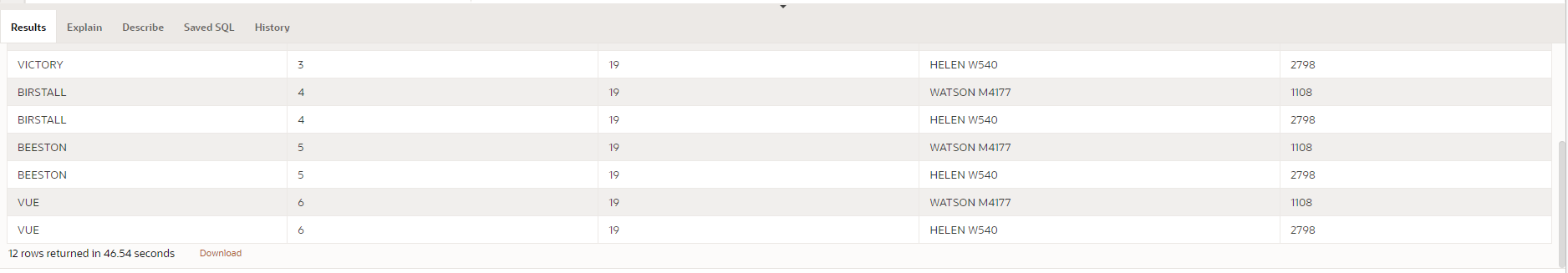
where T1.name=T2.name

and TS1.client\_id =C1.client\_id

and TS2.client\_id=C2.client\_id)

**DATA MART**

****



**RELATIONAL DATA MODEL**

select ops$yyang00.Client.Client#, ops$yyang00.Client.name,ops$yyang00.theatre.Theatre#, ops$yyang00.theatre.name name,totalamount

from ops$yyang00.client,ops$yyang00.ticketpurchase,ops$yyang00.performance,ops$yyang00.theatre

where ops$yyang00.ticketpurchase.Per#=ops$yyang00.performance.Per#

and ops$yyang00.client.client#=ops$yyang00.ticketpurchase.client#

and ops$yyang00.theatre.Theatre#=ops$yyang00.performance.Theatre#

and totalamount=

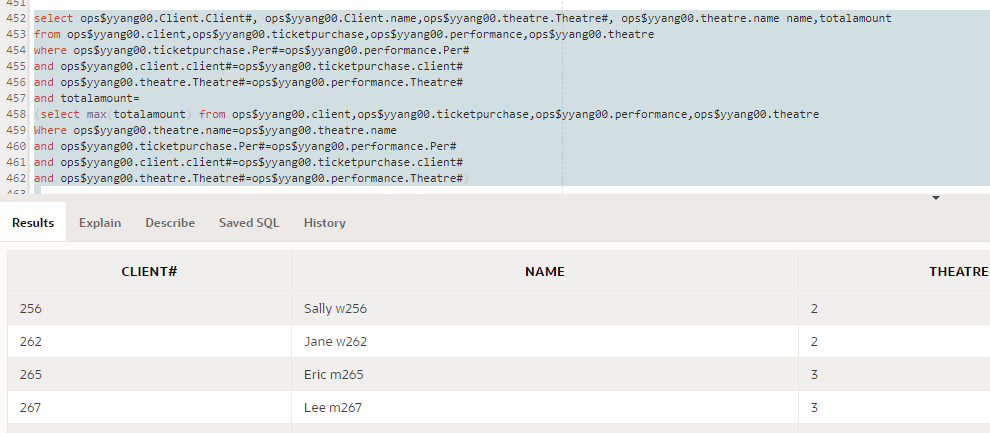
(select max(totalamount) from ops$yyang00.client,ops$yyang00.ticketpurchase,ops$yyang00.performance,ops$yyang00.theatre

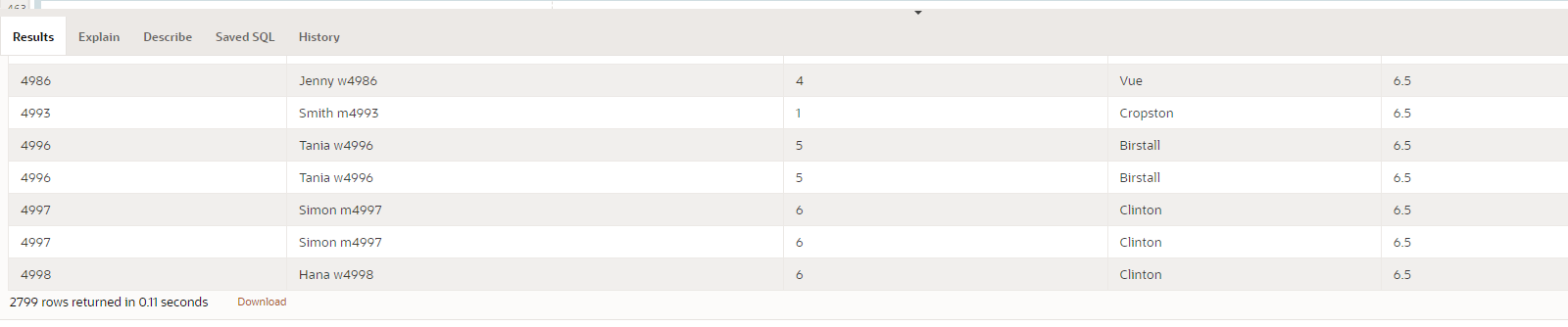
Where ops$yyang00.theatre.name=ops$yyang00.theatre.name

and ops$yyang00.ticketpurchase.Per#=ops$yyang00.performance.Per#

and ops$yyang00.client.client#=ops$yyang00.ticketpurchase.client#

and ops$yyang00.theatre.Theatre#=ops$yyang00.performance.Theatre#)

****

****

**JUSTIFICATION OF THE DATA MART DESIGN.**

The Data Mart is required to assist to have a business intelligence view on the ticket sales. As compared to the Relational Model, The Data Mart focuses on information for Tickets sales.While designing the Star Schema, the major thing to look out for is the Analysis requirement. In the case of Midlands Theatre, the Data mart is expected to analyse the Total Sales Value of each production, The Monthly Sales Value of each theatre and the theatre name (for each theatre) and the names of clients who have the highest spending in that theatre.

Based on this analysis, the required tables are;

1. **Theatre Table** -To get the details of monthly sales value of the theatre and the theatre name for each theatre to know which clients have the highest spending in that theatre.

For the Theatre table, in order to get the total amount from the ticket purchase table we have to link the performance table since there is no direct link with theatre and ticket purchase the performance serves as the link to the two tables.

1. **Production Table**- To get the details of total sales value of each production, for us to get the total amount from the ticket purchase table, we have to link the performance table which serves as a link between the two tables.
2. **Client table**- To get the details of clients with highest spending in each theatre, Client table is directly linked to the Ticket purchase table so we only join client to ticket purchase based on the analysis requirement.
3. **Time Table**- in order to get information on timely basis such as monthly, the table for time has to be created
4. **Ticketsales**- This table contains the main fact of the analysis which is sales value(totalsales). It contains all the primary keys as foreign keys and totalsales as the main requirement for the analysis.

The Data mart makes it easier and more efficient to retrieve data for the required analysis.

**COMPARISM OF THE DATA MART AND RELATIONAL DATA MODEL**

**ANALYSING TOTAL SALES VALUE OF EACH PRODUCTION**

In the Data Mart, to get the total sales of each production the query used in getting the result for is much easier and straightforward since the fact table (Ticketsales) is linked directly to the production table. The Production table is simply joined with the ticket sales table to get the amount. For the relational data model, the query for getting the results seems a lot more complex as there is no link from the production table to the ticket purchase there has to be a join to the performance table. Despite the differences in both queries used, the results from the data mart are exactly the same as the one from the relational data base in regards to the total amount.

**ANALYSING MONTHLY SALE VALUE OF EACH THEATRE**

In the Data Mart, to get the monthly sales value for each theatre we linked the theatre table to the ticket sales and time table. This query meets the analysis requirement and is very straightforward. The Theatre table is joined to the tickets sales to get the totalsales figures and also linked to time table to get the dates for months. For the relational data model, the theatre table is joined to the ticket sales to get the figures for total amount and joined with the performance table to get the dates for monthly values. The query for the relational data base seems a lot more complicated but a review of the output for total amount shows the same figures for both the relational data model and the data mart.

**ANALYSING THE CLIENT’S NAME WITH HIGHEST SPENDING IN EACH THEATRE**

For the Client name with the highest spending in each theatre, to get the get the best result, the focus is on the joining conditions, the first query gets the names of the clients and theatre and total amount sold to them while a subquery is included for the maximum(highest) amount. In the data mart the joining conditions is lesser because the Data mart has been more streamlined to captured details for ticketsales. For the relational data base the joined conditions is quite complex and produces large output.