(BAPT

Data Management and Business Intelligence

Assignment 3

Kaknou Evangelia (BAPTXXX)

Chatzimoschou Angeliki (BAPTXXX)

21 February 2016



# Step 1 Dataset selection

For this assignment the following dataset containing a list of DVD releases was chosen:

<http://www.hometheaterinfo.com/dvdlist.htm>

A MySQL database was installed locally, and the downloaded csv file was imported with the following code:

*/\*Create table to extract data\*/*

create table dvd\_csv\_extract(

DVD\_Title varchar(200),

Studio varchar(200),

Released varchar(200),

Status varchar(200),

Sound varchar(200),

Versions varchar(200),

Price varchar(200),

Rating varchar(200),

Year varchar(200),

Genre varchar(200),

Aspect varchar(200),

UPC varchar(200),

DVD\_ReleaseDate varchar(200),

ID varchar(200),

Timestamp varchar(200));

*/\*Import data from csv\*/*

LOAD DATA INFILE 'dvd\_csv.txt' INTO TABLE dvd\_csv\_extract;

# Step 2 Data quality and staging

The following queries were executed in order to check the quality of data in each column for:

* Null values
* Uniqueness of primary keys
* Frequency distribution of values that are candidates for dimensions

*/\*Check DVD\_TITLE\*/*

select \*

from dvd\_csv\_extract

where DVD\_TITLE is null;

*/\*0 Rows: OK\*/*

*/\*Check Studio\*/*

select \*

from dvd\_csv\_extract

where Studio is null;

*/\*0 Rows: OK\*/*

select Num, count(\*)

from(

select Studio, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by Studio

order by 2 desc)x

group by Num;

*/\*Many studios with just 1 movie\*/*

*/\*Check Released\*/*

select \*

from dvd\_csv\_extract

where Released is null;

*/\*0 Rows: OK\*/*

select Released, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by Released

order by 2 desc;

*/\*46191 rows with no release date\*/*

*/\*Check Status\*/*

select \*

from dvd\_csv\_extract

where status is null;

*/\*0 Rows: OK\*/*

select status, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by status

order by 2 desc;

*/\*OK\*/*

*/\*Check Sound\*/*

select \*

from dvd\_csv\_extract

where Sound is null;

*/\*0 Rows: OK\*/*

select Sound, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by Sound

order by 2 desc;

*/\*OK\*/*

*/\*Check Versions\*/*

select \*

from dvd\_csv\_extract

where Versions is null;

*/\*0 Rows: OK\*/*

select Versions, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by Versions

order by 2 desc;

*/\*OK\*/*

*/\*Check Price\*/*

select \*

from dvd\_csv\_extract

where Price is null;

*/\*0 Rows: OK\*/*

*/\*Check Rating\*/*

select \*

from dvd\_csv\_extract

where Rating is null;

*/\*0 Rows: OK\*/*

select Rating, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by Rating

order by 2 desc;

*/\*OK\*/*

*/\*Check Year\*/*

select \*

from dvd\_csv\_extract

where Year is null;

*/\*0 Rows: OK\*/*

select Year, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by Year

order by 2 desc;

*/\*2 character values 'UNK', 'VAR'\*/*

*/\*Check Genre\*/*

select \*

from dvd\_csv\_extract

where Genre is null;

*/\*0 Rows: OK\*/*

select Genre, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by Genre

order by 2 desc;

*/\*OK\*/*

*/\*Check Aspect\*/*

select \*

from dvd\_csv\_extract

where Aspect is null;

*/\*0 Rows: OK\*/*

select Aspect, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by Aspect

order by 2 desc;

*/\*OK\*/*

*/\*Check DVD\_ReleaseDate\*/*

select \*

from dvd\_csv\_extract

where DVD\_ReleaseDate is null;

*/\*0 Rows: OK\*/*

select DVD\_ReleaseDate, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by DVD\_ReleaseDate

order by 2 desc;

*/\*46191 rows with no release date\*/*

*/\*Check UPC\*/*

select \*

from dvd\_csv\_extract

where UPC is null;

*/\*0 Rows: OK\*/*

select UPC, count(\*) as Num

from dvd\_csv\_extract

group by UPC

having count(\*)>1;

*/\*This column is not unique and cannot be a primary key - column will be dropped\*/*

*/\*Check ID\*/*

select \*

from dvd\_csv\_extract

where ID is null;

*/\*0 Rows: OK\*/*

select ID, count(\*) as Num

from dvd\_csv\_extract

group by ID

having count(\*)>1;

*/\*This column is unique and can be a primary key\*/*

*/\*Create staging table\*/*

DROP TABLE IF EXISTS DVD\_staging;

create table DVD\_staging as

select

DVD\_Title,

a.Studio,

case when b.Num <10 then 'Other' else a.Studio end as StudioGrp,

STR\_TO\_DATE(case when Released = '' then '1900-01-01 00:00:00' else Released end, '%Y-%m-%d %H:%i:%s') as Released,

Status,

Sound,

Versions,

cast(Price as decimal(10,2)) as Price,

Rating,

cast(case when Year in ('UNK', 'VAR') then '1900' else Year end as UNSIGNED) as Year,

Genre,

Aspect,

STR\_TO\_DATE(DVD\_ReleaseDate, '%Y-%m-%d %H:%i:%s') as DVD\_ReleaseDate,

cast(ID as unsigned) as ID

from dvd\_csv\_extract a

left outer join (

select Studio, count(distinct DVD\_TITLE) as Num

from dvd\_csv\_extract

group by Studio

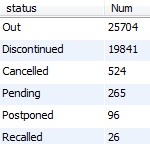
order by 2 desc) b

on b.Studio = a.Studio

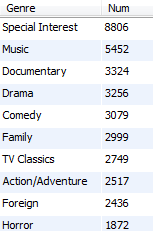
# Step 3 Star schema design

For the star schema the dimensions selected are the following:

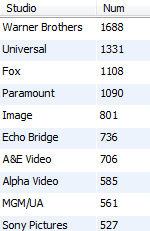
* status\_dim describing the release status of the dvd



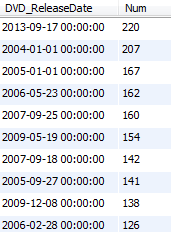
* genre\_dim describing the genre of the movie



* studio\_dim describing the studio that produced the movie

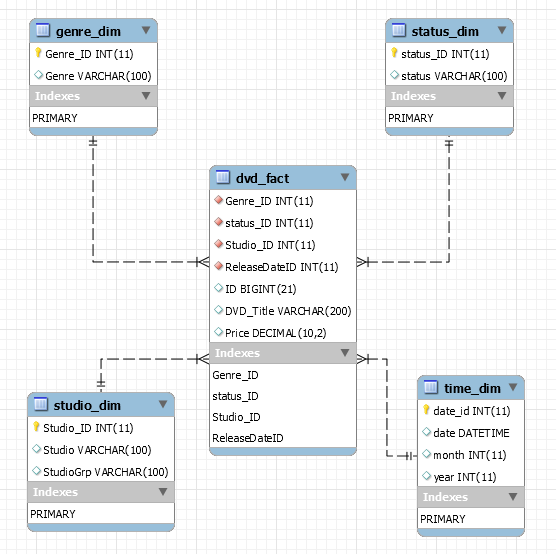


* time\_dim



The granularity of the fact table is one row per dvd with column price as a measure.

The following diagram was generated with EER diagram of MySQL Workbench:



# Step 4 Schema creation and loading

The following queries were executed to create the star schema:

*/\* Studio Dimension\*/*

CREATE TABLE `musicdb`.`studio\_dim` (

`Studio\_ID` INT NOT NULL AUTO\_INCREMENT,

`Studio` VARCHAR(100) NULL,

`StudioGrp` VARCHAR(100) NULL,

PRIMARY KEY (`Studio\_ID`));

*/\* Genre Dimension\*/*

CREATE TABLE `musicdb`.`genre\_dim` (

`Genre\_ID` INT NOT NULL AUTO\_INCREMENT,

`Genre` VARCHAR(100) NULL,

PRIMARY KEY (`Genre\_ID`));

*/\* Status Dimension\*/*

CREATE TABLE `musicdb`.`status\_dim` (

`status\_ID` INT NOT NULL AUTO\_INCREMENT,

`status` VARCHAR(100) NULL,

PRIMARY KEY (`status\_ID`));

*/\* Time Dimension\*/*

CREATE TABLE `musicdb`.`time\_dim` (

`date\_id` INT NOT NULL AUTO\_INCREMENT,

`date` datetime NULL,

`month` int NULL,

`year` int NULL,

PRIMARY KEY (`date\_id`));

*/\*Fact table\*/*

CREATE TABLE `dvd\_fact` (

`Genre\_ID` int(11) NOT NULL DEFAULT '0',

`status\_ID` int(11) NOT NULL DEFAULT '0',

`Studio\_ID` int(11) NOT NULL DEFAULT '0',

`ReleaseDateID` int(11) NOT NULL DEFAULT '0',

`ID` bigint(21) unsigned DEFAULT NULL,

`DVD\_Title` varchar(200) DEFAULT NULL,

`Price` decimal(10,2) DEFAULT NULL,

FOREIGN KEY (Genre\_ID) REFERENCES genre\_dim(Genre\_ID),

FOREIGN KEY (status\_ID) REFERENCES status\_dim(status\_ID),

FOREIGN KEY (Studio\_ID) REFERENCES studio\_dim(Studio\_ID),

FOREIGN KEY (ReleaseDateID) REFERENCES time\_dim(date\_id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

The following queries were executed to load the star schema:

*/\* Studio Dimension\*/*

insert into studio\_dim (Studio,StudioGrp)

select distinct Studio,StudioGrp

from dvd\_staging

where Studio not in (

select distinct Studio

from studio\_dim);

*/\* Genre Dimension\*/*

insert into genre\_dim (genre)

select distinct genre

from dvd\_staging

where genre not in (

select distinct genre

from genre\_dim);

*/\* Status Dimension\*/*

insert into status\_dim (status)

select distinct status

from dvd\_staging

where status not in (

select distinct status

from status\_dim);

*/\* Time Dimension\*/*

insert into time\_dim (date,month,Year)

select distinct DVD\_ReleaseDate, month(DVD\_ReleaseDate),year(DVD\_ReleaseDate)

from dvd\_staging

where DVD\_ReleaseDate not in (

select distinct date

from time\_dim);

*/\*Fact table\*/*

delete from DVD\_fact where ID in (select distinct ID from dvd\_staging);

insert into DVD\_fact

select distinct

b.Genre\_ID,

c.status\_ID,

d.Studio\_ID,

t.date\_id as ReleaseDateID,

a.ID,

a.DVD\_Title,

a.Price

from dvd\_staging a

inner join genre\_dim b

on b.genre = a.genre

inner join status\_dim c

on c.status = a.status

inner join studio\_dim d

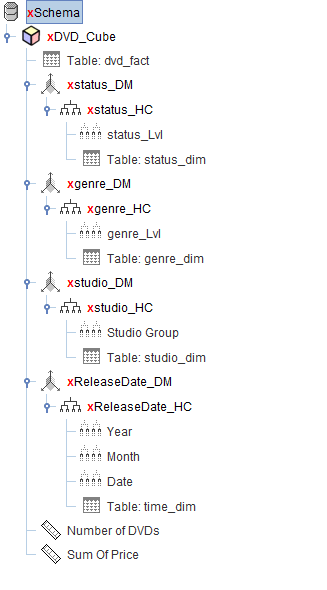
on d.studio = a.studio

inner join time\_dim t

on t.date = a.dvd\_releasedate

# Step 5 Mondrian XML Schema

The following schema was created in Schema Workbench:



<Schema name="DVD">

<Cube name="DVD\_Cube" visible="true" cache="true" enabled="true">

<Table name="dvd\_fact">

</Table>

<Dimension type="StandardDimension" visible="true" foreignKey="status\_ID" name="status\_DM">

<Hierarchy name="status\_HC" visible="true" hasAll="true" allMemberName="All status" allLevelName="All status lvl">

<Table name="status\_dim" alias="">

</Table>

<Level name="status\_Lvl" visible="true" table="status\_dim" column="status\_ID" nameColumn="status" uniqueMembers="false">

</Level>

</Hierarchy>

</Dimension>

<Dimension type="StandardDimension" visible="true" foreignKey="Genre\_ID" name="genre\_DM">

<Hierarchy name="genre\_HC" visible="true" hasAll="true" allMemberName="All genres" allLevelName="All genres lvl">

<Table name="genre\_dim" alias="">

</Table>

<Level name="genre\_Lvl" visible="true" table="genre\_dim" column="Genre\_ID" nameColumn="Genre" uniqueMembers="false">

</Level>

</Hierarchy>

</Dimension>

<Dimension type="StandardDimension" visible="true" foreignKey="Studio\_ID" name="studio\_DM">

<Hierarchy name="studio\_HC" visible="true" hasAll="true" allMemberName="All studios" allLevelName="All studio Lvls" primaryKey="Studio\_ID" primaryKeyTable="studio\_dim">

<Table name="studio\_dim" alias="">

</Table>

<Level name="Studio Group" visible="true" table="studio\_dim" column="Studio\_ID" nameColumn="StudioGrp" uniqueMembers="false">

</Level>

</Hierarchy>

</Dimension>

<Dimension type="StandardDimension" visible="true" foreignKey="ReleaseDateID" name="ReleaseDate\_DM">

<Hierarchy name="ReleaseDate\_HC" visible="true" hasAll="true" allMemberName="All Dates" allLevelName="All Date Levels" primaryKey="date\_id" primaryKeyTable="time\_dim">

<Table name="time\_dim" alias="">

</Table>

<Level name="Year" visible="true" table="time\_dim" column="year" nameColumn="year" uniqueMembers="false" levelType="Regular">

</Level>

<Level name="Month" visible="true" table="time\_dim" column="month" nameColumn="month" uniqueMembers="false" levelType="Regular">

</Level>

<Level name="Date" visible="true" table="time\_dim" column="date" nameColumn="date" uniqueMembers="false" levelType="Regular">

</Level>

</Hierarchy>

</Dimension>

<Measure name="Number of DVDs" column="ID" aggregator="count" visible="true">

</Measure>

<Measure name="Sum Of Price" column="Price" aggregator="sum" visible="true">

</Measure>

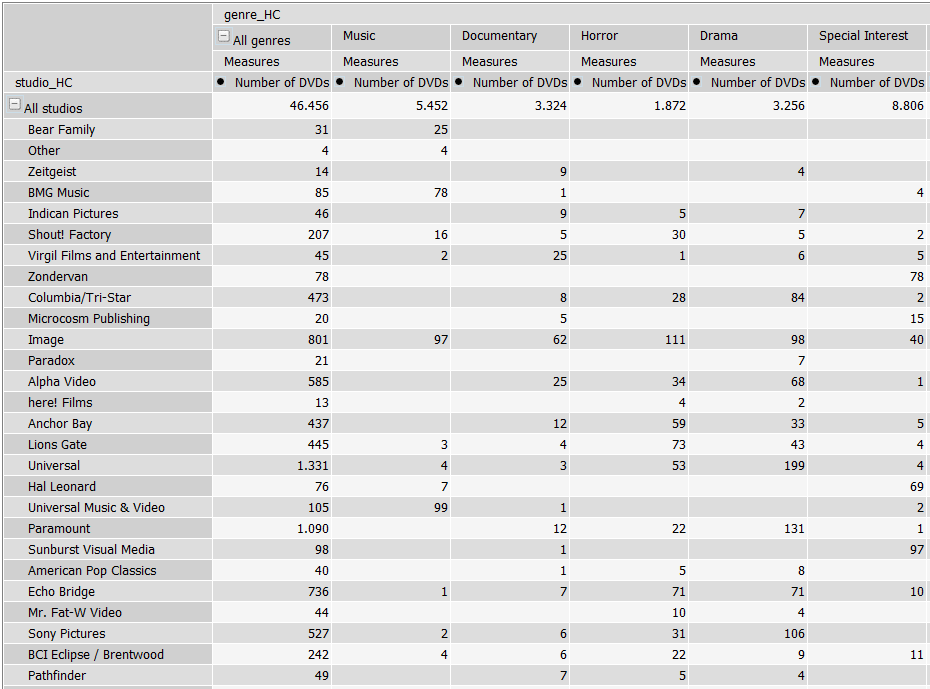
</Cube>

</Schema>

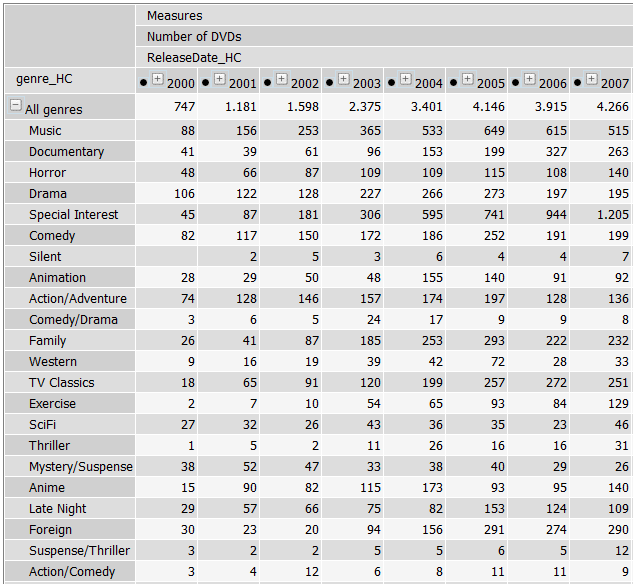
# Step 6 Generation of reports

The following reports were generated using JPivot:

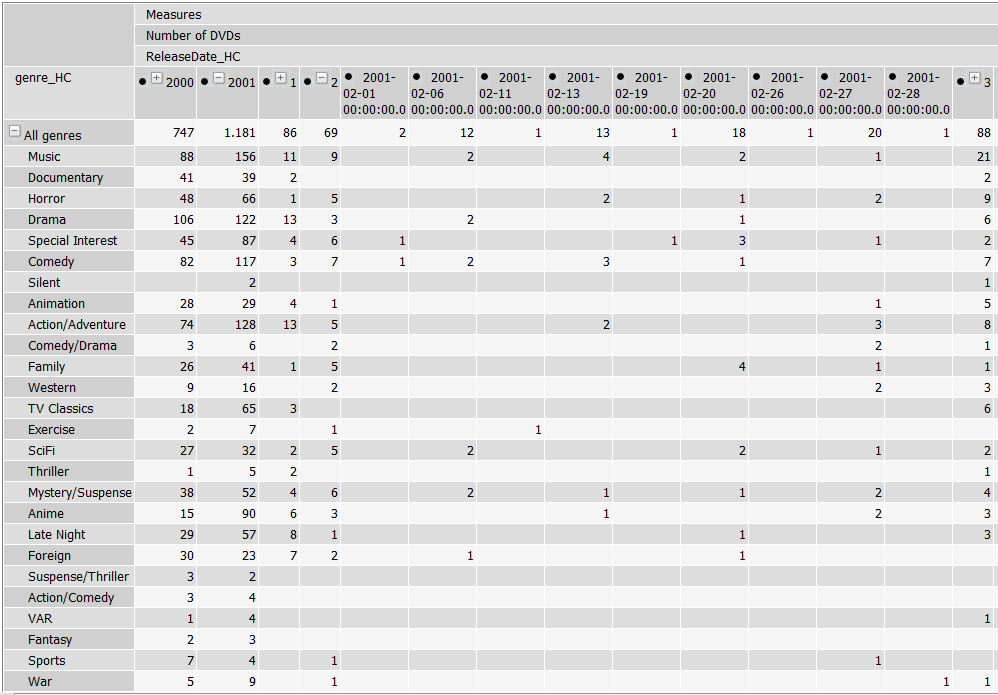
## Report1: Number of DVDs by studio and genre



## Report 2: Number of DVDs per genre and date



### Drill down to time hierarchy:



## Report 3: Sum of price per status, studio and release year

