**1.INTRODUCTION**

**1.1 OBJECTIVE OF THE PROJECT**

The objective of this project is to implement a real time musical database related to Rock music and their respective musical instruments. This program is used to computerise all the data related to music keeping in mind the requirements of the user. The database is created to provide a wide range of data values like Artist, Song, Album, Track, Genre, Label, Musical instruments and Marketing.

The information related to the above specified data values have to be entered and a relationship is established between these. As and when new entry is found in real world it’s populated into database.

**1.2 FEATURES OF THE PROJECT**

A MUSIC database facilitating the objective of the project contains following different tables to satisfy the need of the project. Tables considered in the implementation of the MUSIC database are described below:

**1.2.1 LABEL**

**1.2.2 GENRE**

**1.2.3 ALBUM**

**1.2.4 ARTIST\_GROUP**

**1.2.5 TRACK**

.**1.2.6 SONG**

**1.2.7MARKETING**

**1.2.8MUSICAL\_INSTRUMENTS**

**CHAPTER 2**

1. **System Analysis and specifications**

System Analysis refers into the process of examining a situation with the intent of improving it through better procedures and methods. System Analysis is the process of planning a new System to either replace or complement an existing system. But before any planning is done the old system must be thoroughly understood and the requirements determined. System Analysis, is therefore, the process of gathering and interpreting facts, diagnosing problems and using the information to re-comment improvements in the System. Or in other words, System Analysis means a detailed explanation or description. Before computerized a system under consideration, it has to be analyzed. We need to study how it functions currently, what are the problems, and what are the requirements that the proposed system should meet.

System Analysis is conducted with the following objectives in mind:

1. Identify the customer’s need.
2. Evaluate the system concept for feasibility.
3. Perform economic and technical analysis.
4. Allocate functions to hardware, software people, database and other system elements.
5. Establish cost and schedule constraints.
6. Create a system definition that forms the foundation for all the subsequent engineering work.

**2.1 Requirement Analysis**

* + 1. **Performance Requirement**

The following performance characteristics should be taken care of while developing the system:

* **User friendliness:** The system should be easy to learn and understand so

that **new** user can also use the system effectively, without any difficulty.

**bd14868_ User satisfaction:** The system should meet user expectations.

**bd14868_ Response time:** The response time of all the operations should be low. This can be made possible by careful programming.

**bd14868_ Error handling:** Response to user errors and the undesired situations should be taken care of to ensure that the system operates without halting.

**bd14868_ Safety:** The system should be able to avoid o tackle catastrophic behavior.

**bd14868_ Robustness:** The system should recover from undesired events without human intervention.

* + 1. **System Requirement**

SOFTWARE REQUIREMENT

* + Microsoft Visual Studio 2008
  + Microsoft SQL Server 2005
  + Window XP service Pack 2.

HARDWARE REQUIREMENT

* Pentium IV processor
* 512 RAM
* 80 GB HARD DISK
  + 1. **Acceptance criteria**

The following acceptance criteria were established for the evaluation of the

new system:

* **User friendliness**: - The system should meet user needs and should be easy to learn and use.
* **Modularity**: - The system should have relatively independent and single function parts.
* **Maintainability**: - The system should be such that future maintenance and enhancements times and efforts are reduced.
* **Timeliness**: - The system should operate well under normal, peak and recovery conditions.
* The system developed should be accurate and hence reliable i.e. the error rate should be minimized and the outputs should be consistent and correct.
* Both the execution time and response time should be negligibly low.
* The system should be efficient i.e. the resources utilization should be optimal.
* The system should have scope to forsee modifications and enhancements i.e. it should be able to cope with the changes in future technology.

**Specifications of our project:**

This is the table which gives key information about the Recording company, features and the Genre\_style of the artists. This table also contains some key information which identifies the musical database uniquely.

This table contains the Style , Summary and Information of the artist’s Genre. This table gives a detailed account about the genre present in the album of the artists. Style describes the various style present inside the albums. Where as Summary and Information provides the detailed description of it.

This table contains the details of the individual artists and the bands. It contains the information like the Band\_name, Title, Duration, Rating, Review, Type and the Release\_date of the Album. So this table contains all the details related to a particular album

This table contains the details about the different Artists or Groups present in the Albums. It also contains the information about the Artists, year\_founded, origin, biography and the style of the artist. This table is very helpful to know all the elaborated details about them.

Every Album contains tracks and so do every track has some information related to them like the length of the track,size of the track(in mb),respective track\_num and album\_name and frequency of the track being composed

This table is mainly packed with the album\_name ,it’s recording date ,date of song composition and the song name.

In the complete process of album creation ,marketing it also plays a major role where the profit is dependent on how well we provide publicity and intial composition of song. Here in this table it has specific album name ,intial programmed cost,cost of programmed cd in market ,net profit gained,number of sales done and the unique code num provided which is helpfull in marketing album.

# **The music composition is cannot be possible without the musical instruments,so here in this table it consists of name of instrument used , musical instrument type and the brief explanation about respective musical instrument that is being used.**

**3.3 .State**

LABEL Table

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Rec \_ company | Feature | Genre\_style |
| Iron Maiden | EMI | Pioneers of…….. | Heavy Metal |
| Metallica | Elektra | Is the eponymous… | Hard Rock |
| Green Day | Reprise | Green Day is an… | Punk Rock |
| 30 Seconds to Mars | Virgin | American rock band… | Alternative Rock |
| Linkin Park | WB | Agoura Hills,… | Rap Rock |
| Creed | Wind Up | The band has…. | Post grunge |
| Cold Play | EMI | A Rush of Blood… | Alternative Rock |
| Akon | SRC | Aliaune Badara…. | R & B |

GENRE Table

|  |  |  |
| --- | --- | --- |
| Style | Summary | Information |
| Heavy Metal | The bands that created heavy me. | The heavy metal type of music.. |
| Hard Rock | It is typified by….. | Hard rock is a form of loud….. |
| Alternative Rock | Alternative rock consists of vari.. | Alternative music was generally.. |
| Rap Rock | Rap rock,a genre fusing….. | Rap rock basically fused…. |
| Pop Rock | Pop punk is a fusion music…. | Pop punk, a type of music,which |
| Punk Rock | Punk rock is a rock music genre.. | They created fast, hard-edged … |
| Post grunge | Post-grunge is a subgenre…. | The term post-grunge was coined |
| R & B | Rhythm and blues…… | The term was originally used…. |

ALBUM Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| title | Band\_name | duration | rating | review | type | Release\_date |
| 21st Century | Green Day | 00:69:16 | 2.77 | Critical response to.. | Musically,21st Century…… | 15-may-2009 |
| A Beautiful Lie | 30 Seconds to mars | 00:56:12 | 3.30 | hailing from Southern… | personal and less cerebral.. | 16-Aug-2005 |
| A Thousand Suns | Linkin Park | 00:47:56 | 3.35 | An extremely ... | A Thousand Suns is…. | 14-sep-2010 |
| Appetite for Destruction | Guns N Roses | 00:53:48 | 4.11 | It is great on so…. | Appetite for Destruction is | 21-jul-1987 |
| Human Clay | Creed | 00:56:18 | 3.90 | Human Clay was ranked.. | Human Clay is the second.. | 03-may-1993 |
| Master of Puppets | Metallica | 00:54:45 | 4.40 | The first track,battery | Master of Puppets is the. | 03-mar-1986 |
| Smack That | Akon | 00:16:05 | 4.00 | A remix of "Smack tha.. | "Smack That" is the first… | 28-sep-2006 |
| Parachutes | Cold Play | 00:41:48 | 3.7 | Dont Panic-Dont Panic.. | Parachutes is the debut… | 26-aug-2002 |
| Dance of Death | Iron Maiden | 00:67:59 | 3.87 | Iron Maiden has been…. | Dance of Death is…. | 02-sep-2003 |

ARTIST\_GROUP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| name | Year\_founded | origin | biography | style |
| Iron Maiden | 1975 | Leyton in east London, | The band was basically………. | Heavy Metal |
| 30 Seconds to Mars | 1998 | Los Angeles American | 30 Seconds to Mars is an …….. | Hard rock,Post grunge' |
| Green Day | 1987 | Gilman Street in Berkeley,California. | Green Day was originally part…. | Punk Rock |
| Linkin Park | 1996 | Agoura Hills, California. | Linkin Park’s foundation was…. | Rap Rock |
| Guns N Roses | 1985 | Los Angeles, California | The group was formed……… | hard rock |
| Cold Play | 1996 | London | Chris Martin and Jonny Buckland | Alternative Rock |
| Akon | 2003 | America | Akon alleged three years…………… | R & B |
| Creed | 1996 | Florida State | Founding members,vocalist……. | Post grunge |

TRACK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Track\_num | Size\_mb | length | frequency | Album\_name |
| im\_track01 | 3mb | 00:01:46 | 128kbps | Killers |
| im\_track02 | 5mb | 00:04:18 | 128kbps | Killers |
| im\_track03 | 4mb | 00:03:22 | 128kbps | Killers |
| im\_track04 | 3mb | 00:02:54 | 128kbps | Killers |
| im\_track06 | 4mb | 00:03:53 | 128kbps | Killers |
| im\_track07 | 3mb | 00:01:46 | 128kbps | Killers |
| im\_track08 | 3mb | 00:02:33 | 128kbps | Killers |
| im\_track09 | 6mb | 00:06:11 | 128kbps | Killers |

SONG

|  |  |  |  |
| --- | --- | --- | --- |
| Song\_name | Recording\_date | Date\_composed | Album\_name |
| im\_kl\_The Ides Of March | 10-jan-1981 | 01-nov-1980 | Killers |
| im\_ps\_Aces High | 10-jul-1984 | 01-sep-1983 | Powerslave |
| im\_dod\_Wildest Dreams | 10-aug-2003 | 01-dec-2002 | Dance of Death |
| gd\_21stCB\_21 guns | 10-apr-2009 | 01-jan-2008 | 21st Century Breakdown |
| gd\_ai\_"American Idiot | 26-mar-2004 | 18-apr-2003 | American Idiot |
| lp\_ts\_Burning in the ski | 14-sep-2010 | 10-aug-2010 | A Thousand Suns |
| lp\_ht\_In the End | 24-oct-2000 | 10-aug-2000 | Hybrid Theory |
| lp\_met\_Numb | 29-jun-2004 | 25-mar-2003 | Meteora |

MARKETING

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Code\_num | Pro\_cost | Cost\_cd | Net\_profit | sales | Album\_name |
| abc-001 | $ 3.4 millons | $ 5 | around $ 40 million | around 10 million cds | 21st Century Breakdown |
| abc-002 | $ 4.4 millon | $ 6 | around $ 25 million | around 18 million cds | American Idiot |
| abc-003 | $ 4.6 millons | $ 10 | around $ 15 million | around 20 million cds | A Beautiful Lie |
| abc-004 | $ 7 millon | $ 10 | around $ 15 million | around 25 million cds | A Thousand Suns |
| abc-005 | $ 5 millon | $ 6 | around $ 30 million | around 25 million cds | Hybrid Theory |
| abc-006 | $ 6 millon | $ 7 | around $ 27 million | around 23 million cds | Meteora |
| abc-007 | $ 7 millon | $ 7 | around $ 17 million | around 13 million cds | Minutes |
| abc-008 | $ 8 millon | $ 8 | around $ 17 million | around 20 million cds | Reanimation |

MUSICAL\_INSTRUMENTS

|  |  |  |
| --- | --- | --- |
| name | type | Information |
| Electric Guitar | string instrument | Electric guitar was first……… |
| Acoustic Guitar | string instrument | Even though the Guitars…….. |
| Drum | Percussion | Drums were one of the………. |
| Piano | String | The piano is a musical………. |
| Keyboard | Piano | A musical keyboard is the…… |
| Flute | Wind instrument | Flute is one of the……… |
| Violin | String instrument | Violin was basically developed… |

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

**4.1. Introduction to sql**

The SQL(Structured Query Language) and relational database systems based on it are one of the most important foundation technologies in the computer industry. Over the last two decades, SQL has grown from its first commercial use into a computer product and services market segment worth tens of billions of dollars per year, and SQL stands today as the standard computer database language. Literally hundreds of database products now support SQL, running on computer systems from mainframes to personal computers and even handheld devices. An official international SQL standard has been adopted and expanded twice. Virtually every major enterprise software product relies on SQL for its data management, and SQL is at the core of the database products from Microsoft, Oracle, and IBM, the three largest software companies in the world. SQL is also at the heart of open-source database products that are helping to fuel the popularity of Linux and the open-source movement. From its obscure beginnings as an IBM research project, SQL has leaped to prominence as both an important computer technology and a powerful market force.

SQL is a database sublanguage for querying and modifying relational databases. It was developed by IBM Research in the mid 70's and standardized by ANSI in 1986.

* SQL is used to control all of the functions that a DBMS provides for its users, including:
* **Data definition.** SQL lets a user define the structure and organization of the stored data and relationships among the stored data items.
* **Data retrieval.** SQL allows a user or an application program to retrieve stored data from the database and use it.
* **Data manipulation.** SQL allows a user or an application program to update the database by adding new data, removing old data, and modifying previously stored data.
* **Access control.** SQL can be used to restrict a user’s ability to retrieve, add, and modify data, protecting stored data against unauthorized access.
* **Data sharing.** SQL is used to coordinate data sharing by concurrent users, ensuring that they do not interfere with one another.
* **Data integrity.** SQL defines integrity constraints in the database, protecting it from corruption due to inconsistent updates or system failures.

SQL is a comprehensive language for controlling and interacting with a database management system.

* **Clauses**, which are constituent components of statements and queries. (In some cases, these are optional.
* **Expressions**, which can produce either scalars values or tables consisting of columns and rows of data.
* **Predicates**, which specify conditions that can be evaluated to SQL three-valued logic (3VL) or Boolean (true/false/unknown) truth values and which are used to limit the effects of statements and queries, or to change program flow.
* **Queries**, which retrieve the data based on specific criteria. This is the most important element of **SQL**.
* **Statements**, which may have a persistent effect on schemata and data, or which may control transactions, program flow, connections, sessions, or diagnostics.
  + SQL statements also include the semicolon (";") statement terminator. Though not required on every platform, it is defined as a standard part of the SQL grammar.
* Insignificant whitespace is generally ignored in SQL statements and queries, making it easier to format SQL code for readability.

**CLAUSES:**

The FROM clause which indicates the table(s) from which data is to be retrieved. The FROM clause can include optional JOIN subclauses to specify the rules for joining tables.

* The WHERE clause includes a comparison predicate, which restricts the rows returned by the query. The WHERE clause eliminates all rows from the result set for which the comparison predicate does not evaluate to True.
* The GROUP BY clause is used to project rows having common values into a smaller set of rows. GROUP BY is often used in conjunction with SQL aggregation functions or to eliminate duplicate rows from a result set. The WHERE clause is applied before the GROUP BY clause.
* The HAVING clause includes a predicate used to filter rows resulting from the GROUP BY clause. Because it acts on the results of the GROUP BY clause, aggregation functions can be used in the HAVING clause predicate.
* The ORDER BY clause identifies which columns are used to sort the resulting data, and in which direction they should be sorted (options are ascending or descending). Without an ORDER BY clause, the order of rows returned by an SQL query is undefined.

There are 3 basic categories of SQL Statements:

* SQL-Data Statements-- query and modify tables and columns
  + SELECT Statements -- query tables and views in the database
  + INSERT Statements -- add rows to tables
  + UPDATE Statements -- modify columns in table rows
  + DELETE Statements -- remove rows from tables
* SQL- Transaction Statements -- control transactions
  + COMMIT Statements -- commit the current transaction
  + ROLLBACK Statements-- roll back the current transaction
* SQL- Schema Statements -- maintain schema (catalog)
  + CREATE TABLE Statements -- create tables
  + CREATE VIEW Statements -- create views
  + DROP TABLE Statements -- drop tables
  + DROP VIEW Statements-- drop views
  + GRANT Statements -- grant privileges on tables and views to other users
  + REVOKE Statements-- revoke privileges on tables and views from other users.

**Language Structure**

SQL is a keyword based language. Each statement begins with a unique keyword. SQL statements consist of clauses which begin with a keyword. SQL syntax is not case sensitive. The other lexical elements of SQL statements are:

* names -- names of database elements: tables, columns, views, users, schemas; names must begin with a letter (a - z) and may contain digits (0 - 9) and underscore (\_)
* literals -- quoted strings, numeric values, datetime values
* delimiters -- + - , ( ) = < > <= >= <> . \* / || ? ;

Basic database objects (tables, views) can optionally be qualified by schema name. A dot".", separates qualifiers:

Example:

schema-name . table-name

**4.2. Queries to be executed and results to be added**

**4.2.1. RELATIONAL ALGEBRA QUERIES**

* This relational query is used to select the ALBUMS which is having the rating value above 4.

σrating>4(ALBUM)

* This query is used to select the “Title of the ALBUM” and “Track number” from the TRACK table whose length is greater than 00:05:00 minutes.

πtitle,track\_num (σlength>00:05:00(ALBUM⋈album\_name=titleTRACK))

* This query is used to select the columns “code\_num” and “album\_name” from tables “MARKETING” and “ALBUM” whose production cost is greater than $ 8 millions and cost of each cd greater than $ 5 .

πcode\_num,album\_name (σ(production\_cost>8)**AND(**cost\_cd<5**)**MARKETING)

* This query is used to select “album\_name” from ALBUM table, marketing “code\_num” from MARKETING table and LABEL name whose net\_profit is greater than or equal to $ 15 million or production cost greater than $ 6 million ,and production company is ‘EMI’

**T1** MARKETING⋈album\_name=title ALBUM

**T2**  ((net\_profit>=15**OR**production\_cost>6)**AND(**rec\_company=’EMI’)) (T1⋈band\_name=name LABEL)

**T3** π album\_name,code\_num,net\_profit(T2)

**T3** contains the required result.

* This query is used to obtain the “name” and “type” of musical instruments.

π name,type(MUSICAL\_INSTRUMENTS)

* This query is used to get the get the genre\_style from table “LABEL” and name from “ARTIST\_GROUP” which are founded after 1987.

**T1**  ARTIST\_GROUP⋈(style=genre\_style)LABEL

π genre\_style,name(σ(year\_founded>1987)**T1**)

**4.2.2. SQL QUERIES**

CREATE COMMAND :-

* create table LABEL( name varchar(30) not null,record\_comapny varchar(30), feature varchar(400),Genre\_style varchar(40),primary key(name),foreign key(Genre\_style) references Genre(style));
* create table LABEL( name varchar(30) not null,record\_comapny varchar(30),feature varchar(400),Genre\_style varchar(40),primary key(name), foreign key(Genre\_style) references Genre(style));
* create table ALBUM( band\_name varchar(30),title varchar(30) not null,duration varchar(10),rating number(10,5),review varchar(500),type varchar(500),release\_date date,primary key(title),foreign key(band\_name) references label(name));
* create table ARTIST\_GROUP( name varchar(30) not null,year\_fo varchar(200) not null,origin varchar(200), biography varchar(300),style varchar(30),primary key(name), foreign key(style) REFERENCES GENRE (style));
* create table SONG( name varchar(50) not null, recording\_date date,date\_composed date,album\_name varchar(40),primary key(name),foreign key(album\_name) references album(title), foreign key(band\_name) references label(name));
* create table TRACK( track\_num varchar(20) not null,size\_mb int not null,length varchar(10),frequency int,channels int, album\_name varchar(40),primary key(track\_num),foreign key(name) REFERENCES SONG(name),foreign key(album\_name) references album(title),foreign key(band\_name) references label(name));
* create table MARKETING( code\_num varchar(10) not null,production\_cost varchar(30) not null, cost\_cd varchar(30),net\_profit varchar(30),sales varchar(30), album\_name varchar(30),primary key(code\_num),foreign key(album\_name) references album(title));
* create table MUSICAL\_INST( name varchar(40) not null,type varchar(40) not null,info varchar(500),price varchar(10),primary key(name));

INSERT COMMAND :-

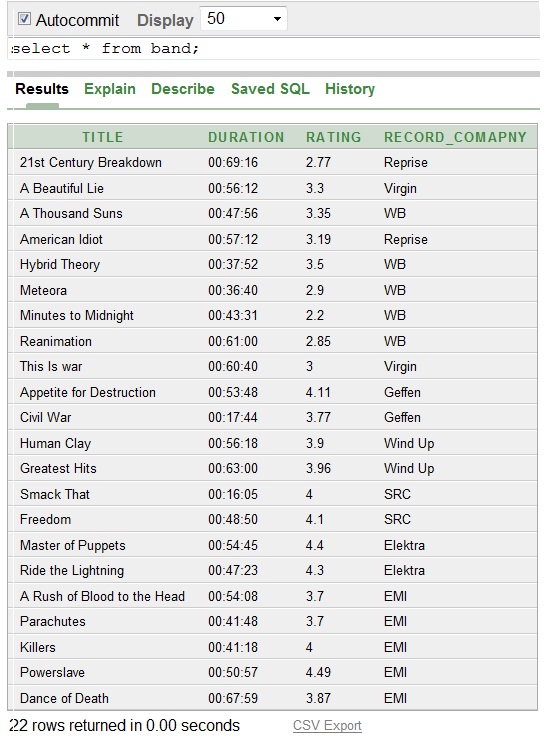
* insert into LABEL values ('Iron Maiden','EMI','Iron Maiden are an English heavy metal band.Pioneers of genre-New wave of British Heavy Metal.Its sound and music has inspired countless other bands in the heavy metal and rock genre.Metallica drummer "Lars Ulrich" said "Iron Maiden have been the blueprint for everything that we have ever wanted to do.I have always had an incredible amount of respect and admiration for them."','Heavy Metal');
* insert into GENRE values ('Alternative Rock','Alternative rock consists of various subgenres that have emerged from the independent music scene since the 1980s, such as grunge, gothic rock and indie rock. These genres are unified by this style of music.','Alternative music was generally originated in the late 1980s and it basically included many genres and it brought new dimension to the music industry. Anything which is different is categorized under the alternate type of music.');
* insert into ALBUM values ('Linkin Park','A Thousand Suns','00:47:56','3.35','An extremely well-crafted rock album from a band whose (self-directed) anger is, for once, well placed.A Thousand Suns isnt quite the return to the magic of debut album Hybrid Theory, as it has been billed in some quarters, but neither is it a straight re-enactment of the turgid attempt at a stadium-filler that was 2007s Minutes to Midnight.Musically, A Thousand Suns is more intricately put together than any of the bands previous releases, even the ultimately superior Hybrid Theory.','A Thousand Suns is the fourth studio album by American rock band Linkin Park.The style of the album has been compared to the music of Pink Floyd and contains the high pitched sonar "ping" from Echoes at the end of "Robot Boy" and continuing into "Jornada del Muerto".A Thousand Suns is a concept album dealing with human fears such as nuclear warfare.It has also been stated that the album honors the hip-hop group Public Enemy.','14-sep-2010');
* insert into ARTIST\_GROUP values('Creed','1996','Founding members,vocalist Scott Stapp and guitarist Mark Tremonti had been classmates in high school and friends at Florida State University.They found their love for music and laid the foundation.','Creed is an American rock band formed in 1995 in Tallahassee, Florida. Becoming popular in the late 1990s and early 2000s, the band has released three consecutive multi-platinum albums, one of which has been certified diamond.','Post grunge');
* insert into SONG values ('30stm\_abl\_A Beautiful Lie','29-aug-2005','23-feb-2003','A Beautiful Lie');
* insert into TRACK values ('gng\_track03','3mb','00:03:17','128kbps','Appetite for Destruction');
* insert into MARKETING values ('abc-015','$ 10 millon','$ 9','aronnd $ 21 million','around 18 million cds','Freedom');
* insert into MUSICAL\_INST values ('Electric Guitar','string instrument','Electric guitar was first used by Gage Brewer in 1930s. An electric guitar is a guitar that uses the principle of electro magnetic pickup to convert vibrations of its strings into electric signals.They rely on electronic amplification to make their sound audible.','6700rs');

ALTER COMMAND :-

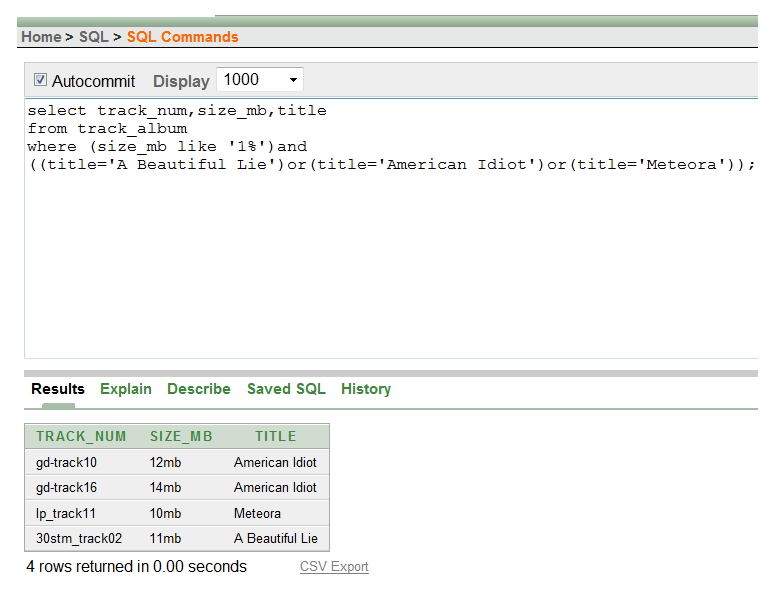
* alter table SONG drop column BAND\_NAME;
* alter table SONG add music\_inst varchar(30);
* alter table SONG add foreign key(music\_inst) references MUSICAL\_INST(name);
* alter table SONG drop column music\_inst;

VIEW COMMAND :-

* create view band as select title,duration,rating,record\_comapny from album,label where band\_name=name;
* select \* from band;

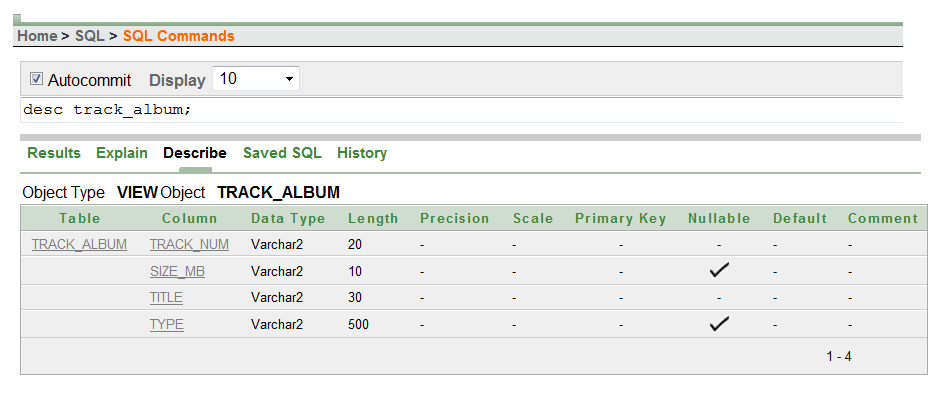


* create view track\_album as select track\_num,size\_mb,title,type from album,track where album\_name=title;
* select track\_num,size\_mb,title from track\_album where (size\_mb like '1%')and((title='A Beautiful Lie')or(title='American Idiot')or(title='Meteora'));

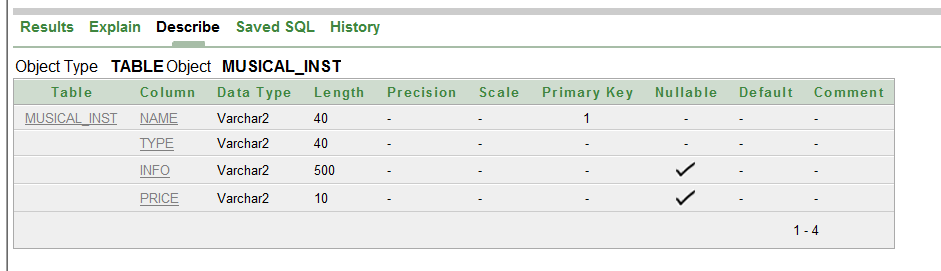


DESC COMMAND :-

* desc track\_album;



* desc musical\_inst;



DROP COMMAND :-

* drop view band;
* drop table label;

DELETE COMMAND :-

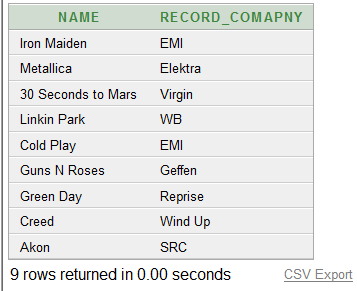
* delete from marketing where net\_profit like ‘%2%’;
* delete from album where rating>2.5;

UPDATE COMMAND :-

* update track\_album set size\_mb='8mb' where track\_num='gd-track05';
* update marketing set net\_profit=’around $ 30 million’ where code\_num=’abc006’;

JOIN AND OUTER JOINS IN SQL :-

* select name,record\_comapny from label JOIN genre on genre\_style=style;

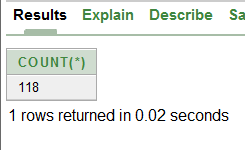


SELECT AND AGGREGATE FUNCTIONS :-

* select sum(rating),max(rating),min(rating),avg(rating) from album;



* select count(\*) from track,album where album\_name=title ;

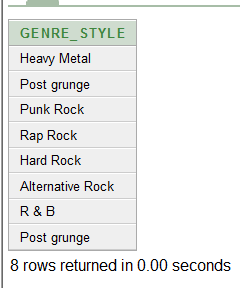


ASSERTIONS :-

* create assertion rating\_al check(exists(select \* from album where rating>3.5));

ALIASING :-

* select style as genre\_style from artist\_group;



ORDER BY :-

* select A.style,G.style from artist\_group A,genre G where a.style=g.style order by A.style DESC,G.style;



* select title,band\_name,size\_mb from album,track where (album\_name=title) and (size\_mb like '\_mb') order by band\_name desc,title; 

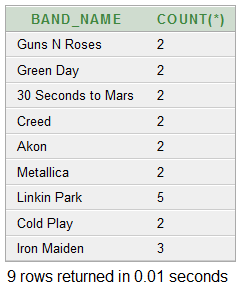


GROUP BY:-

* select album\_name,count(\*) from song group by album\_name having count(\*)>=8;

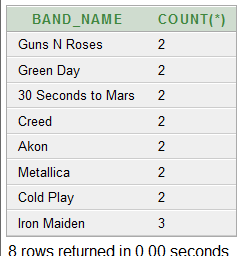


* select band\_name,count(\*) from label,album where name=band\_name group by band\_name;



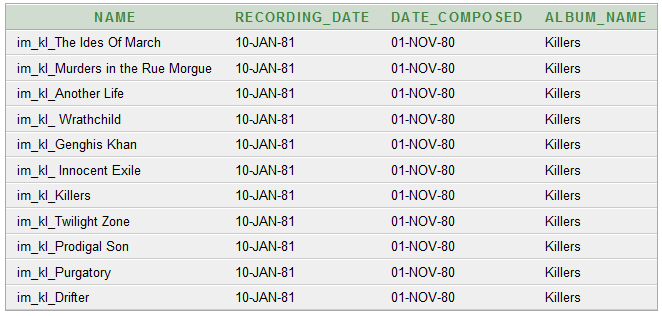
HAVING :-

* select band\_name,count(\*) from label,album where name=band\_name group by band\_name having count(\*)<5;

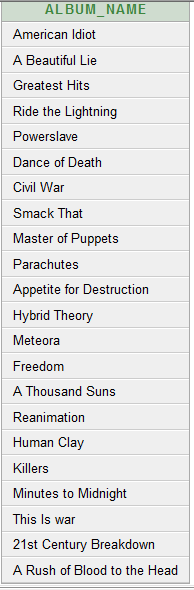


NESTED QUERIES :-

* select \* from song where recording\_date=(select min(recording\_date) from song);

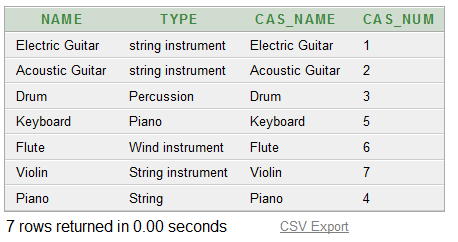


* select distinct album\_name from song where album\_name in (select album\_name from marketing,album where album\_name=title);

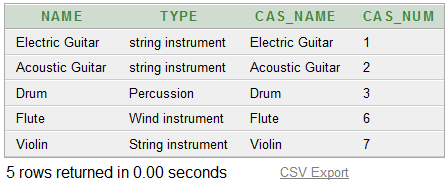


CASCADE OPTION :-

* create table music\_cas (cas\_name varchar(20) references musical\_inst(name) on delete cascade,cas\_num int primary key);
* select name,type,cas\_name,cas\_num from musical\_inst,music\_cas where name=cas\_name;



* delete from musical\_inst where name='Piano' or name='Keyboard';
* select name,type,cas\_name,cas\_num from musical\_inst,music\_cas where name=cas\_name;



**CHAPTER 5**

**SYSTEM TESTING AND RESULTS**

**UNIT TESTING**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL\_NO | COMMAND | ACTUAL RESULT | EXPECTED RESULT | REMARK |
| 1 | CREATE | 8 | 8 | CREATED |
| 2 | INSERT | 8 | 8 | INSERTED |
| 3 | ALTER | 4 | 4 | ALTERED |
| 4 | DROP | 2 | 2 | DROPPED |
| 5 | DELETE | 2 | 2 | DELETED |
| 6 | SELECT | 11 | 11 | SELECTED |
| 7 | DESC | 2 | 2 | DESCRIBED |
| 8 | ALIAS | 1 | 1 | SUCCESS |
| 9 | ORDER BY | 2 | 2 | ORDERED |
| 10 | GROUP BY | 2 | 2 | GROUPED |
| 11 | HAVING | 2 | 2 | SUCCESS |
| 12 | UPDATE | 2 | 2 | UPDATED |
| 13 | NESTED QUERY | 2 | 2 | SUCCESS |
| 14 | AGGREGATE | More than 3 | More than  3 | ALL CREATED |
| 15 | VIEW | 3 | 3 | CREATED |
|  |  |  |  |  |

The Music and Musical instruments Database has been created .We have tried Alter,drop, delete etc.., using Oracle 10g Express Edition and it is found to be executed successfully. It is working properly for each individual operations.

The database is populated successfully and it’s found to be working properly in all aspects even after integrating the tables, views etc… flawlessly. So it’s clear that there is no problem after integrating the individual aspects of the database.

**CHAPTER 6**

**NORMALIZATION OF THE DESIGN**

**ARTIST\_GROUP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **Year\_**  **founded** | **origin** | **biography** | **style** |

FUNCTONAL DEPENDENCIES

Name { year\_founded,origin,biography,style}

I NORMAL FORM

In above relation ARTIST\_GROUP all the attributes are atomic , there are no multivalued attributes and composite attributes. Hence it is in I NORMAL FORM.

II NORMAL FORM

In the above relation ARTIST\_GROUP all non prime attributes of it are fully functionally dependent on the primary key name. Hence it is in II NORMAL FORM.

III NORMAL FORM

As the above relation is in II NORMAL FORM and there are no transitive dependencies. Hence it is in III NORMAL FORM.

**ALBUM**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **title** | **band\_name** | **duration** | **rating** | **review** | **type** | **release\_date** |

FUNCTONAL DEPENDENCIES

Title {band\_name,duration,rating,review,type,release\_date}

I NORMAL FORM

In above relation **ALBUM** , all the attributes are atomic , there are no multivalued attributes and composite attributes. Hence it is in I NORMAL FORM.

II NORMAL FORM

In the above relation **ALBUM**, all non prime attributes of it are fully functionally dependent on the primary key Ipan. Hence it is in II NORMAL FORM.

III NORMAL FORM

As the above relation **ALBUM** is in II NORMAL FORM and there are no transitive dependencies. Hence it is in III NORMAL FORM.

**TRACK**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| track\_num | Size\_mb | length | frequency | Album\_name | Song\_name |

I NORMAL FORM

In above relation **TRACK** , all the attributes are atomic , there are no multivalued attributes and composite attributes. Hence it is in I NORMAL FORM.

II NORMAL FORM

In the above relation TRACK all non prime attributes of it are not fully functionally dependent on the primary key . Hence it is normalised in to II NORMAL FORM.

By spliting the table in to two functional dependancies.

**TRACK**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| track\_num | size\_mb | length | frequency | Album\_name |

FUNCTONAL DEPENDENCIES

Track\_num {size\_mb,length,frequency,Album\_name }

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Size\_mb | length | frequency | Album\_name | Song\_name |

Song\_name {size\_mb,length,frequency,album\_name}

III NORMAL FORM

As the above relation is in II NORMAL FORM and there are no transitive dependencies. Hence it is in III NORMAL FORM.

**CHAPTER 7**

**ANALYSIS OF DATAPHOR**

**Dataphor** is an open-source truly Relational database management system (RDBMS) and its accompanying user interface technologies, which together are designed to provide highly declarative software application development. The Dataphor Server is a virtual, or federated, DBMS, meaning that it utilizes other database engines for storage.

The stated purpose of Dataphor is to attempt to raise the bar of automation when building and maintaining complex software applications. Originally referred to as a framework, Dataphor provides more of a software development platform, complete with its own programming and user interface paradigms.

Dataphor is broadly divided into two components: the Dataphor Server, and the Dataphor Frontend. The purpose of the Dataphor Server is to provide a standardized language and runtime for the definition, manipulation, and integrity of application data. The Frontend is concerned with the dynamic derivation of user interfaces and the presentation thereof in either the Windows or Web thin client.

Dataphor does not employ SQL as its primary database language since SQL purportedly violates important principles of the relational model. Dataphor's D4 language is based on the principles of Christopher J Date's and Hugh Darwen's Tutorial D, but with a Pascal-like imperative syntax.

Though Dataphor espouses to be truly relational, it does incorporate the concept of NULLs as found in SQL, which many claim to be contraindicated by the Relational Model. NULLs and the matter of managing missing information, however, continue to be debated.

In addition to the data management focus of the Dataphor Server, Dataphor includes tools which allow the presentation of user interfaces through Windows and Web "thin" clients. Dataphor takes advantage of the relational inference capabilities of the Dataphor compiler in order to allow complete GUI forms to be derived directly from the data model. The unique aspect of Dataphor's user interface "derivation" is that it may be based on any relational expression (query) rather than merely base tables.

**7.1 FEATURES**

• Centralized and extensible data dictionary

• Dynamic user interface derivation

• Declarative environment for business rules

• Multiple clients from a single description

• Rich data manipulation algebra

**7.2 TECHNOLOGY**

• Database Management System (DMBS) written in C# for the version 2.0 .NET Framework

• Compiler and runtime for D4 language

• Integrated Development Environment (IDE) for:

Editing and executing D4

Browsing the data dictionary catalog

Analyzing execution plans

Visually building forms, or customizing derived forms

• Windows Forms based thin client

• ASP.NET based web server thin "client"

• Libraries to access SQL Server, Oracle, DB2, and a few other data sources

• Libraries for standard programming tasks as well as web related tasks.

**7.3 GOALS**

Dataphor is a database application development platform that automates many of the tasks involved in the creation and maintenance of database applications. The Dataphor Server provides a logical abstraction of database management systems, while Dataphor Frontend provides a logical abstraction of user interface definitions. Alphora Dataphor allows developers to focus on the creation of the core data model and underlying process logic. Based on the given data model, Dataphor dynamically derives user interfaces and enforces business rules. This data-centric approach, which is enabled by Dataphor’s query processor and data dictionary, enables developers to:

1) Focus on what it is they are modeling rather than the implementation details;

2) Shield themselves from the limitations and diversity of existing data sources; and

3) Produce more flexible applications in less time.

**7.4 LAYERS**

The Dataphor approach consists of describing applications in three main layers: Physical, Logical, and Presentation. The Physical Layer is responsible for managing the persistent data. The Presentation Layer is responsible for managing the interaction with user. The Logical Layer is responsible for providing the actual description of the application. The Logical layer is described in terms of constructs within a virtual database server. To the extent possible, Dataphor tries to automate the implementation of the Physical and Presentation layers by inferring information from the Logical.

**7.5 Dataphor Server**

The Dataphor Server has several components including:

* Call level interface - session management, process scheduler, etc.
* Data Dictionary Catalog - containing all of the Tables, Views, Operators, Constraints, References, and other schema objects.
* D4 Scanner, Parser, Emitter, and Compiler.
* D4 Runtime - including relational, and scalar processing
* Storage Integration layer - real-time translation to various dialects of SQL

**7.6 Frontend Library**

The Dataphor Frontend library provides for the delivery of dynamically derived, or pre-designed static forms. The library is exposed as a standard set of D4 functions (called operators in D4) such as Form('<library>', '<name>') and Derive('<D4 expession>', '<form type>'). The resulting forms are described in an XML dialect called a Dataphor Form Document (DFD). The form description is high-level, consisting of a general description of the user interface aspects as they apply independent of client platform.

**CHAPTER 8**

**8. CONCLUSION AND FUTURE WORK**

By this Project we have created the databaseof Rock Musicand all aspects related to it. Using a database management system (ORACLE 10g) we have implemented according to user’s convenience.

The information about the Artists, Groups, Genre, Label, Tracks, Songs, Marketing and Musical Instruments are loaded into the database and these aspects of the music are related and a relation is established b/w these and hence as and when new entry is found in real world it’s populated into database.

By the analysis of Dataphor, We have learnt some basic knowledge like Features, Compatibility and Applications. And also we have learnt how to work in a team. Consulting people in the Musical field was great and we thank them for their support.

We thank for giving us an opportunity to do this project which helped us a lot in learning database management systems

Hence by this project we have learnt to create, maintain and juggle with the Database.

**FUTURE WORK**:

Having this back-end, in future we can implement a colorful front-end of the same by uising PHP or any other Front-end designing tool . We can implement in

* 1. Online music database which maintains the data about all aspects of the music online and shares the data to the world.
  2. An online music streaming website where an individual can create his own playlist and add his favourite songs to it. He can login to his personal account and view or listen to his most listened songs anytime and anywhere.

**CHAPTER 9**

**REFERENCES**

* **www.dataphor.org(yyuyu)**
* **www.wikepedia.org/wiki/sql()**
* **The Complete Reference** (ISBN 0-07-222559-9, 2004) by James R. Groff and Paul N. Weinberg.
* Elmasri Navathe **'Fundamentals of database systems'.**
* **Principles of Distributed Database Systems,** By -Prentice-Hall
* **Ramprasad K.M**, Senior Software Designer, Hawlett -Packard, Bangalore.