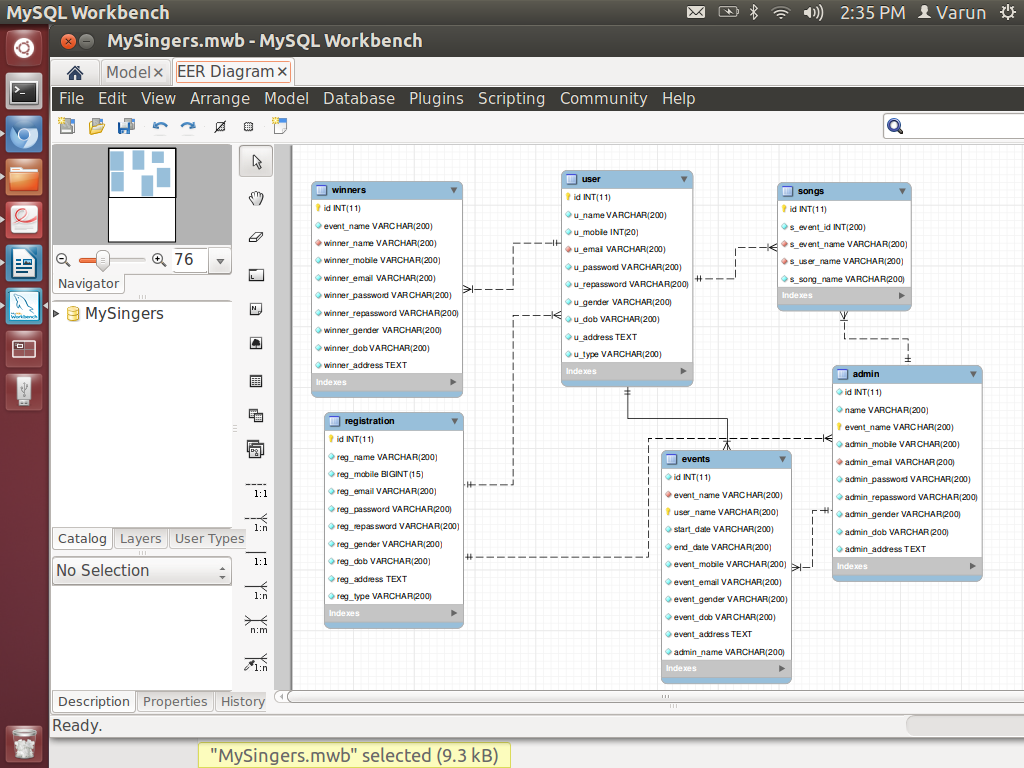
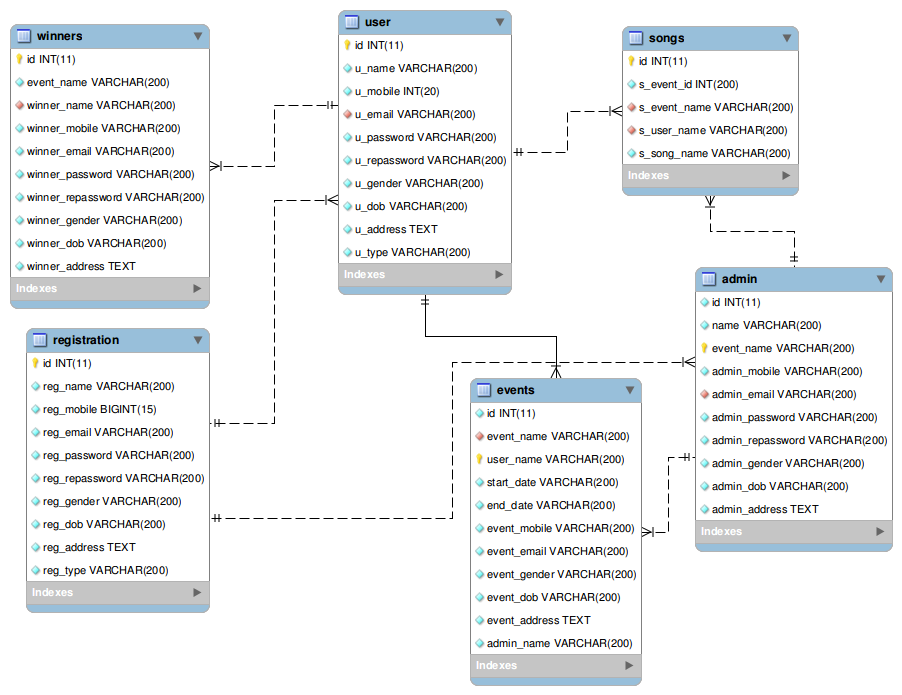
**My Singers**

**EER-Diagram:** As show the EER Diagram as follow.





**EER-Diagram for Registration with Administrator:**

|  |  |
| --- | --- |
| EER-Diagram | Relationship |
| A description... | A registration have many administrators, i.e One to Many relationships. Similarlly A administrator have to register one time. |

**EER-Diagram for Registration with User:**

|  |  |
| --- | --- |
| EER-Diagram | Relationship |
| A description... | A registration have many Users, i.e One to Many relationships. Similarlly A User have to register one time. |

**EER-Diagram for User with Songs:**

|  |  |
| --- | --- |
| EER-Diagram | Relationship |
| A description... | A User have singes many songs, i.e One to Many relationships. Similarlly A Sonss are singing by the many user. |

**EER-Diagram for User with Events:**

|  |  |
| --- | --- |
| EER-Diagram | Relationship |
| A description... | A User have participates in many events, i.e One to Many relationships. Similarlly many events participates by the user. |

**EER-Diagram for User with Winners:**

|  |  |
| --- | --- |
| EER-Diagram | Relationship |
| A description... | A User annaconce the many winners, i.e One to Many relationships. Similarlly many winners have a user. |

**EER-Diagram for Administrator with Events:**

|  |  |
| --- | --- |
| EER-Diagram | Relationship |
| A description... | A Administrator participates the many winners, i.e One to Many relationships. Similarlly many events manged by the administrator. |

**EER-Diagram for Administrator with Songs:**

|  |  |
| --- | --- |
| EER-Diagram | Relationship |
| A description... | A Administrator participates the many Songs, i.e One to Many relationships. Similarlly many Songs manged by the administrator. |

**Metadata Table**

**KEY** or **INDEX** refers to a normal non-unique index. Non-distinct values for the index are allowed, so the index *may* contain rows with identical values in all columns of the index. These indexes don't enforce any restraints on your data so they are used only for making sure certain queries can run quickly.

**UNIQU** refers to an index where all rows of the index must be unique. That is, the same row may not have identical non-NULL values for all columns in this index as another row. As well as being used to speed up queries, UNIQUE indexes can be used to enforce restraints on data, because the database system does not allow this distinct values rule to be broken when inserting or updating data.

**The primary key is used to identify a row of data in a table and the values of a PK must be unique A table can have multiple unique and foreign keys. However, a table can have only one primary key.**

**A foreign key can actually reference a key that is not the primary key of a table. But, a foreign key must reference a *unique* key. A While unique and primary keys both enforce uniqueness on the column(s) of one table.foreign key can hold NULL values.**

**Unique key columns are allowed to hold NULL values. The values in a primary key column, however, can ever be NULL.**

**Registration Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Column Names | Data Type | Keys |
| Registration | id | int(11) | PRIMARY key |
| reg\_name | varchar(200) | Index Key |
| reg\_mobile | bigint(15) |  |
| reg\_email | varchar(200) | Unique Index Key |
| reg\_password | varchar(200) |  |
| reg\_repassword | varchar(200) |  |
| reg\_gender | varchar(200) |  |
| reg\_dob | varchar(200) |  |
| reg\_address | text |  |
| reg\_type | varchar(200) |  |

Table:

CREATE TABLE IF NOT EXISTS `registration` (

`id` int(11) NOT NULL,

`reg\_name` varchar(200) NOT NULL,

`reg\_mobile` bigint(15) NOT NULL,

`reg\_email` varchar(200) NOT NULL,

`reg\_password` varchar(200) NOT NULL,

`reg\_repassword` varchar(200) NOT NULL,

`reg\_gender` varchar(200) NOT NULL,

`reg\_dob` varchar(200) NOT NULL,

`reg\_address` text NOT NULL,

`reg\_type` varchar(200) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `reg\_email` (`reg\_email`),

KEY `reg\_name` (`reg\_name`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Description:

I have to add the primary key to id, unuque key to reg\_email nad key to reg\_name.

**Admin Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Column Names | Data Type | Keys |
| Admin | id | int(11) |  |
| name | varchar(200) |  |
| event\_name | varchar(200) | Primary key,Unique Index Key |
| admin\_mobile | varchar(200) |  |
| admin\_email | varchar(200) | Index key |
| admin\_password | varchar(200) |  |
| admin\_repassword | varchar(200) |  |
| admin\_gender | varchar(200) |  |
| admin\_dob | varchar(200) |  |
| admin\_address | text |  |

Table:

CREATE TABLE IF NOT EXISTS `admin` (

`id` int(11) NOT NULL,

`name` varchar(200) NOT NULL,

`event\_name` varchar(200) NOT NULL,

`admin\_mobile` varchar(200) NOT NULL,

`admin\_email` varchar(200) NOT NULL,

`admin\_password` varchar(200) NOT NULL,

`admin\_repassword` varchar(200) NOT NULL,

`admin\_gender` varchar(200) NOT NULL,

`admin\_dob` varchar(200) NOT NULL,

`admin\_address` text NOT NULL,

PRIMARY KEY (`event\_name`),

KEY `admin\_email` (`admin\_email`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Description: I have added the primary key to event\_name and then key to admin\_email column. The remaining columns are declared string type with NOT NULL.

**User Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Column Names | Data Type | Keys |
| User | id | int(11) | Primary key |
| u\_name | varchar(200) | Index Key |
| u\_mobile | varchar(200) |  |
| u\_email | varchar(200) | Unique Index Key |
| u\_password | varchar(200) |  |
| u\_repassword | varchar(200) |  |
| u\_gender | varchar(200) |  |
| u\_dob | varchar(200) |  |
| u\_address | text |  |

Table:

CREATE TABLE IF NOT EXISTS `user` (

`id` int(11) NOT NULL,

`u\_name` varchar(200) NOT NULL,

`u\_mobile` int(20) NOT NULL,

`u\_email` varchar(200) NOT NULL,

`u\_password` varchar(200) NOT NULL,

`u\_repassword` varchar(200) NOT NULL,

`u\_gender` varchar(200) NOT NULL,

`u\_dob` varchar(200) NOT NULL,

`u\_address` text NOT NULL,

`u\_type` varchar(200) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `u\_email` (`u\_email`),

KEY `u\_name` (`u\_name`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Relationships: In this table have to applyed key to u\_name column and unuque key to e\_email.

A user have only a one e\_email id in this table. Not allowed duplicate values.

**Events Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Column Names | Data Type | Keys |
| Events | id | int(11) | Index key |
| event\_name | varchar(200) | Index key |
| user\_name | varchar(200) | Primary key, Uniue Index |
| start\_date | varchar(200) |  |
| end\_date | varchar(200) |  |
| event\_mobile | varchar(200) |  |
| event\_email | Varchar(200) |  |
| event\_gender | varchar(200) |  |
| event\_dob | varchar(200) |  |
| event\_address | text |  |
| admin\_name | varchar(200) | Index Key |

Table:

CREATE TABLE IF NOT EXISTS `events` (

`id` int(11) NOT NULL,

`event\_name` varchar(200) NOT NULL,

`user\_name` varchar(200) NOT NULL,

`start\_date` varchar(200) NOT NULL,

`end\_date` varchar(200) NOT NULL,

`event\_mobile` varchar(200) NOT NULL,

`event\_email` varchar(200) NOT NULL,

`event\_gender` varchar(200) NOT NULL,

`event\_dob` varchar(200) NOT NULL,

`event\_address` text NOT NULL,

`admin\_name` varchar(200) NOT NULL,

PRIMARY KEY (`user\_name`),

UNIQUE KEY `id` (`id`),

KEY `admin\_name` (`admin\_name`),

KEY `event\_name` (`event\_name`),

KEY `user\_name` (`user\_name`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Relationships: In this table have applyed the key to “event\_name” and “user\_name” columns.

**Songs Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Column Names | Data Type | Keys |
| Songs | id | Int(11) | Primary, Unuque key |
| s\_event\_id | int(20) |  |
| s\_event\_name | varchar(200) | Index key |
| s\_user\_name | varchar(200) | Index key |
| s\_song\_name | varchar(200) | Index key |

Table:

CREATE TABLE IF NOT EXISTS `songs` (

`id` int(11) NOT NULL,

`s\_event\_id` int(200) NOT NULL,

`s\_event\_name` varchar(200) NOT NULL,

`s\_user\_name` varchar(200) NOT NULL,

`s\_song\_name` varchar(200) NOT NULL,

PRIMARY KEY (`id`),

KEY `s\_user\_name` (`s\_user\_name`),

KEY `s\_song\_name` (`s\_song\_name`),

KEY `s\_event\_name` (`s\_event\_name`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Relationships: In this table have to set the primary key to “id” column and then key to “s\_event\_name”, “s\_song\_name” and “s\_user\_name”.

**Winner Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Column Names | Data Type | Keys |
| Winner | id | int(11) | Primary key |
| evemt\_name | varchar(200) |  |
| Winner\_name | varchat(200) | Index key |
| winner\_mobile | varchar(200) |  |
| winner\_email | varchar(200) |  |
| winner\_password | varchar(200) |  |
| winner\_repassword | varchar(200) |  |
| winner\_gender | varchar(200) |  |
| winner\_dob | varchar(200) |  |
| winner\_address | text |  |

Table:

CREATE TABLE IF NOT EXISTS `winners` (

`id` int(11) NOT NULL,

`event\_name` varchar(200) NOT NULL,

`winner\_name` varchar(200) NOT NULL,

`winner\_mobile` varchar(200) NOT NULL,

`winner\_email` varchar(200) NOT NULL,

`winner\_password` varchar(200) NOT NULL,

`winner\_repassword` varchar(200) NOT NULL,

`winner\_gender` varchar(200) NOT NULL,

`winner\_dob` varchar(200) NOT NULL,

`winner\_address` text NOT NULL,

PRIMARY KEY (`id`),

KEY `winner\_name` (`winner\_name`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Relationships: In this table have to set the primary key to “id” column and index key to “winner\_name” column.