# INFO 6210
# Data Management and Database Design

# Kickstarter Crowdfunding Recommendation Engine

# Semester End Project (SQL)


Submission Date: April 26, 2019

**Members:**

Name: Shaishav Shah
NU ID: 001447053

Name: Parth Rana
NU ID: 001417553

Name: Anshul Balamwar
NU ID: 001400625

# Brief about project idea:

Kickstarter is one of the most popular crowdfunding platforms on the internet. The aim of this project is to predict the success or failure of a Kickstarter campaign at launch time.

Crowdfunding is the practice of funding a project or venture by raising monetary contributions from many people. The majority of today's crowdfunding happens online through various websites and one of the most prominent is Kickstarter. The steps to start a Kickstarter project are; start a campaign, set the minimum funding goal, set reward levels, and choose a deadline. The most important aspect to know about launching a Kickstarter project is that if the project falls short of meeting its minimum funding goal, the project will not receive any fund. The projects analyzed in this project fall into one of 14 categories (Art, Comics, Dance, Design, Fashion, Film & Video, Film & amp; Video, Food, Games, Music, Photography, Publishing, Technology, Theater) and 51 subcategories. Only 55% of campaigns reach their funding goal thus it is extremely important for creators to know the factor(s) that might impact the outcome of their project before launch.

This project will take inputs from users using website and machine learning algorithms will provide various prediction / recommendations which are helpful to conduct the crowdfunding project.

Input from the Users on Website / Predictor for ML Algorithm

- Category and Subcategory of Project
- Location of the Project (City and State)
- Goal in Dollars
- Levels, Duration and No. of Update for the Project

# Conceptual Schema:

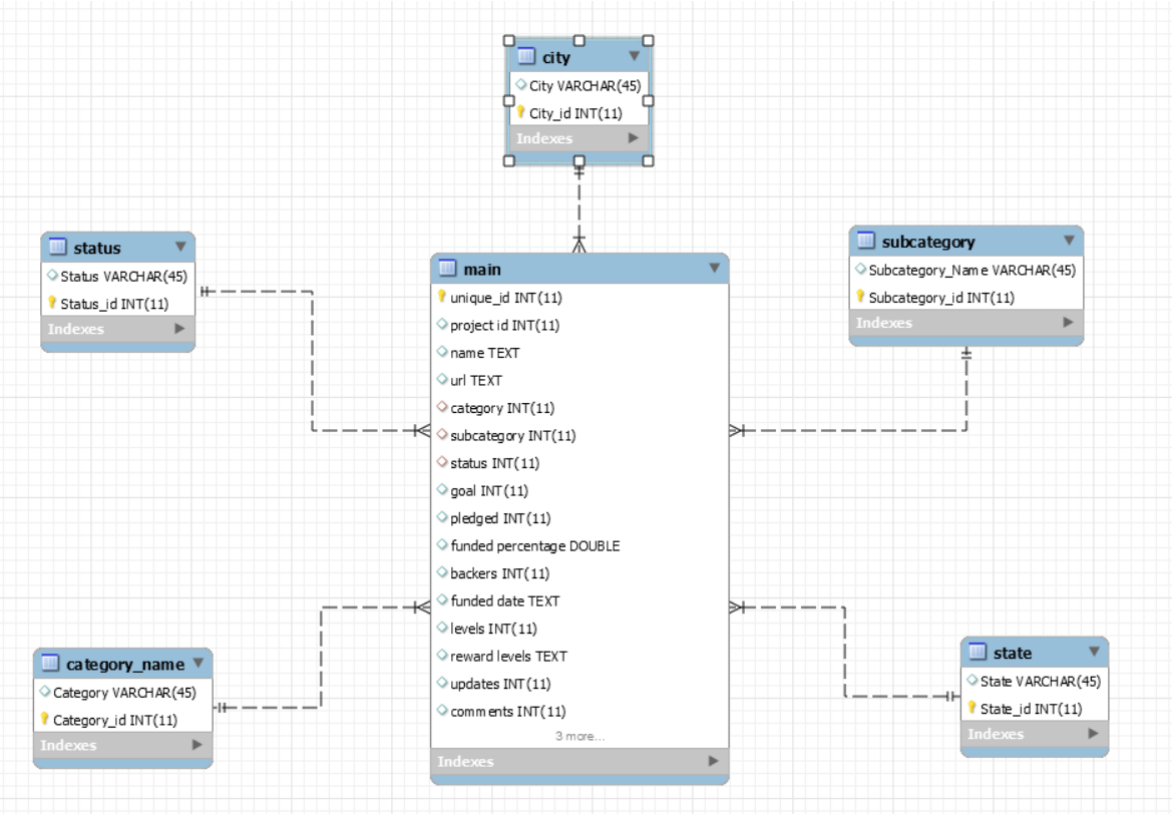
The database consists of 6 tables which are in 3rd normal form. The main table consist of all the project details all the details entered by the user to estimate the success or failure of a project.

The smaller tables consist of the integer values assigned to the categorical variables which are then used to build the random forest method to predict the classification of the project in successful or failed project list.

The data set consist of 46000 rows and 16 attributes or features.

About the table:

1. City: Primary Key: City_id
2. State: Primary Key: State_id
3. Subcategory: Primary Key: Subcategory_id
4. Category: Primary Key: Category_id
5. Status: Primary Key: Status_id
6. Main:
    a. Primary Key: unique_id
    b. Foreign Key: city, state, category, subcategory, status



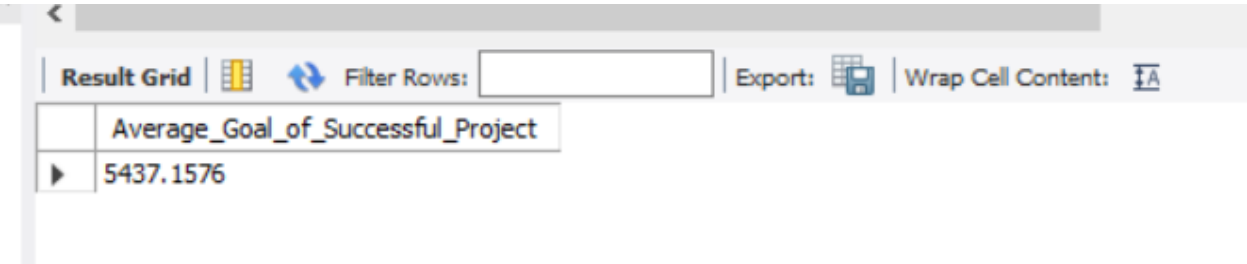
# Exploration of the database: SQL Queries

#1 Average goal of successful and failed projects

Query 1:

```
select avg(goal) as Average_Goal_of_Successful_Project from main
where status = 1;
```

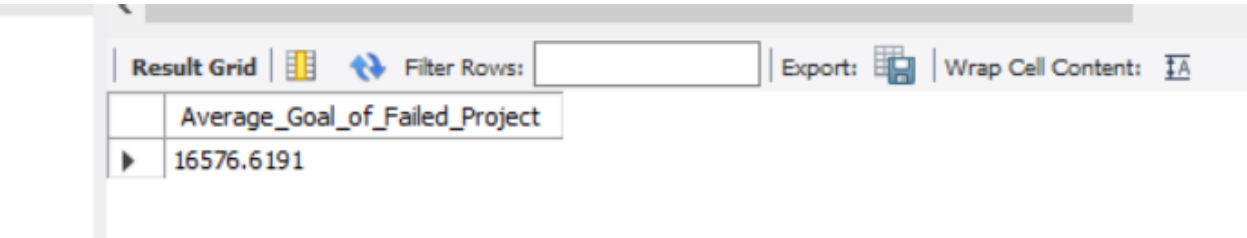
Output:




Query 2:

```
select avg(goal) as Average_Goal_of_Failed_Project from main
where status = 0;
```

Output:

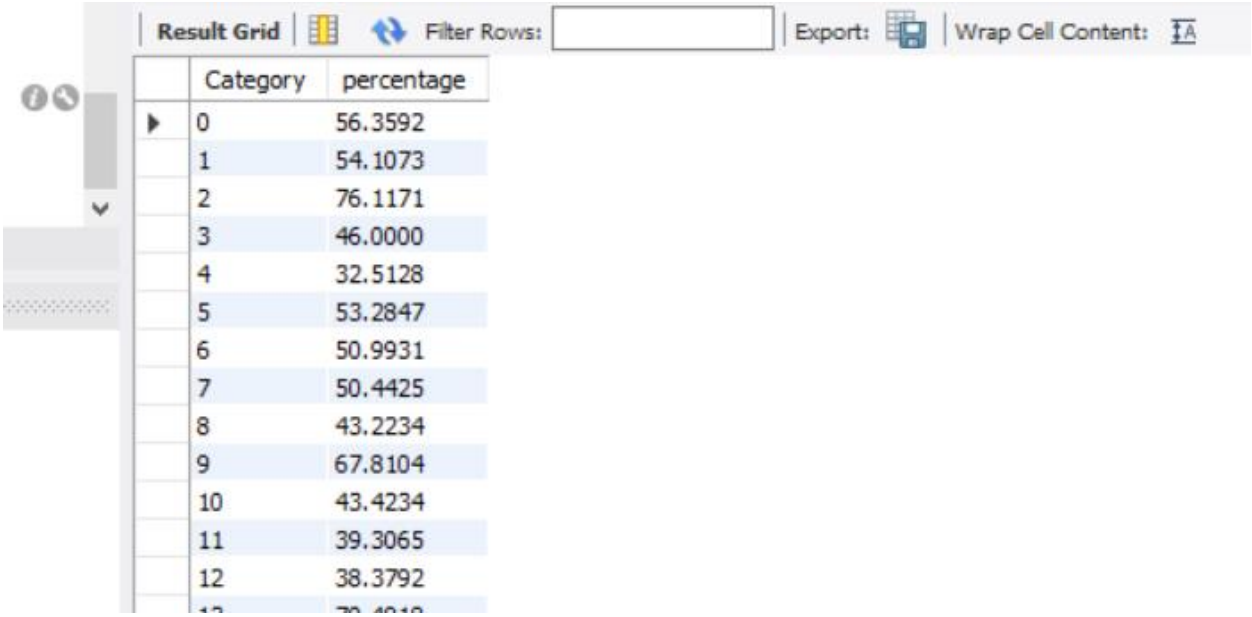



#2 Success Percentage according to category

Query 3:

```
select Category , avg(status)*100 as percentage from main
group by category;
```

Output:



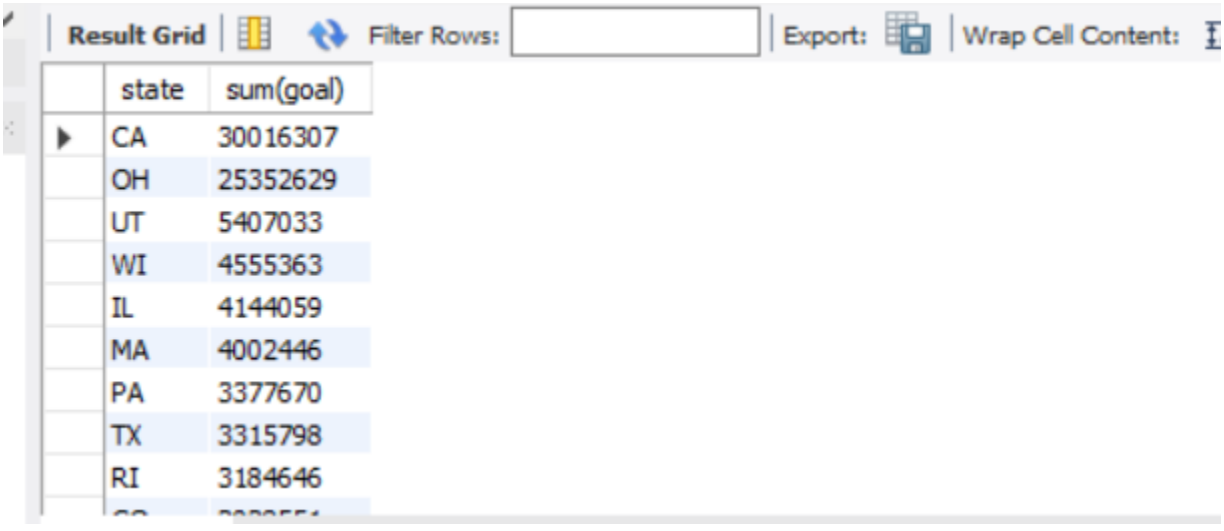
| Category | percentage |
|---|---|
| 0 | 56.3592 |
| 1 | 54.1073 |
| 2 | 76.1171 |
| 3 | 46.0000 |
| 4 | 32.5128 |
| 5 | 53.2847 |
| 6 | 50.9931 |
| 7 | 50.4425 |
| 8 | 43.2234 |
| 9 | 67.8104 |
| 10 | 43.4234 |
| 11 | 39.3065 |
| 12 | 38.3792 |

#3 Total funding in a state

Query 4:

```
select state.state, sum(goal) from main left join state on main.state = state.State_id
where status = 1
group by state
order by sum(goal) desc;
```

Output:



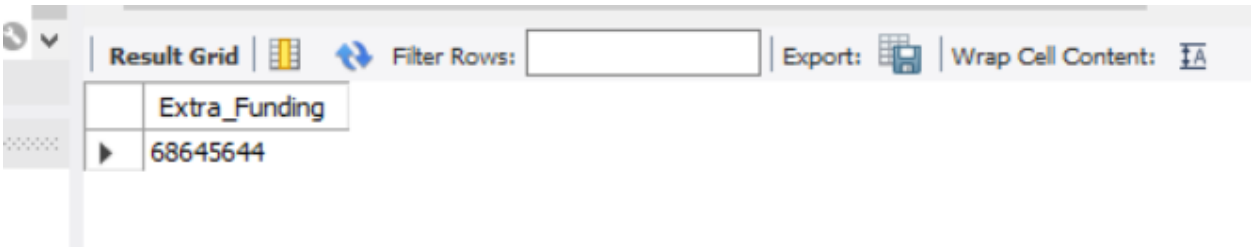
| state | sum(goal) |
|---|---|
| CA | 30016307 |
| OH | 25352629 |
| UT | 5407033 |
| WI | 4555363 |
| IL | 4144059 |
| MA | 4002446 |
| PA | 3377670 |
| TX | 3315798 |
| RI | 3184646 |

#4 Total Extra Funding

Query 5:

```
select sum(pledged-goal) as Extra_Funding from main
where status = 1;
```

Output:




Query 6:

```
select sum(pledged-goal) as Extra_Funding from main
where status = 0;
```

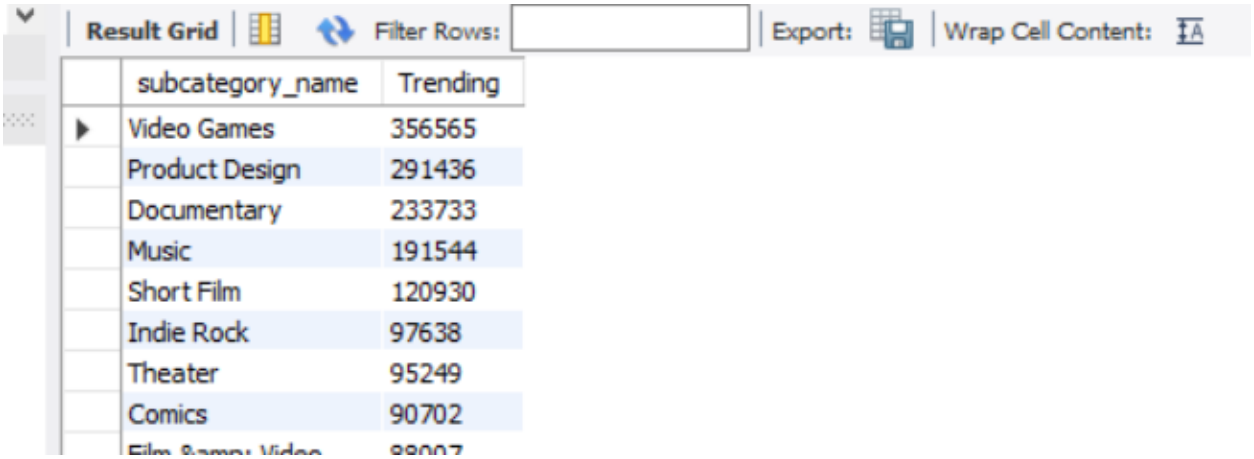
Output:


#5 Trending subcategory

Query 7:

```
select subcategory_name, sum(backers) as Trending
from main left join subcategory on main.subcategory = subcategory.subcategory_id
group by subcategory
order by sum(backers) desc;
```

Output:




| subcategory_name | Trending |
|---|---|
| Video Games | 356565 |
| Product Design | 291436 |
| Documentary | 233733 |
| Music | 191544 |
| Short Film | 120930 |
| Indie Rock | 97638 |
| Theater | 95249 |
| Comics | 90702 |

#6 Average goal of successful and failed projects

Query 8:

```
select avg(goal) from main
where status = 1;
```

Query 9:

```
select avg(goal) from main
where status = 0;
```

Output:


| avg(goal) |
|---|
| 16576.6191 |

#8 Total Projects for success and failure

Query 10:

```
select count(unique_id) from main
where status = 1;
```

Output:


Query 11:

```
select avg(unique_id) from main
where status = 0;
```

Output:


#9 Average Updates

Query 12:

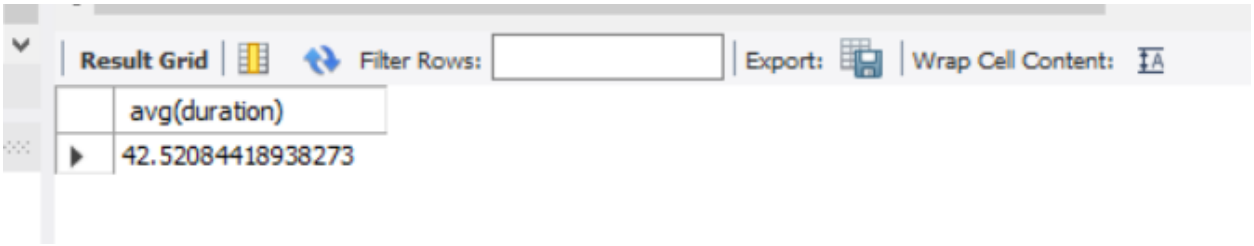
```
select avg(duration) from main
where status = 1;
```

Output:


Query 13:

```
select avg(duration) from main
where status = 0;
```

Output:

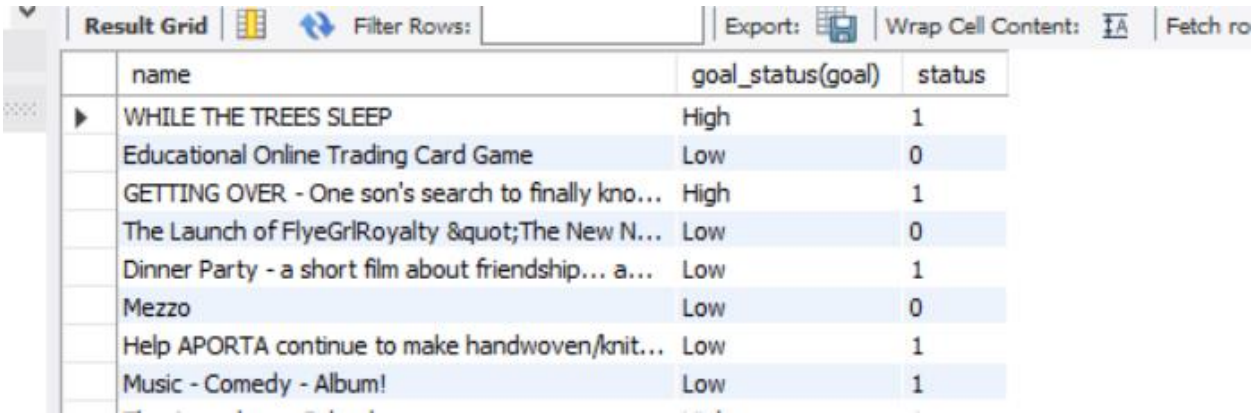



#10

Query 14 with function:

```
select name, backers_amount(backers) from main;
```

Output:

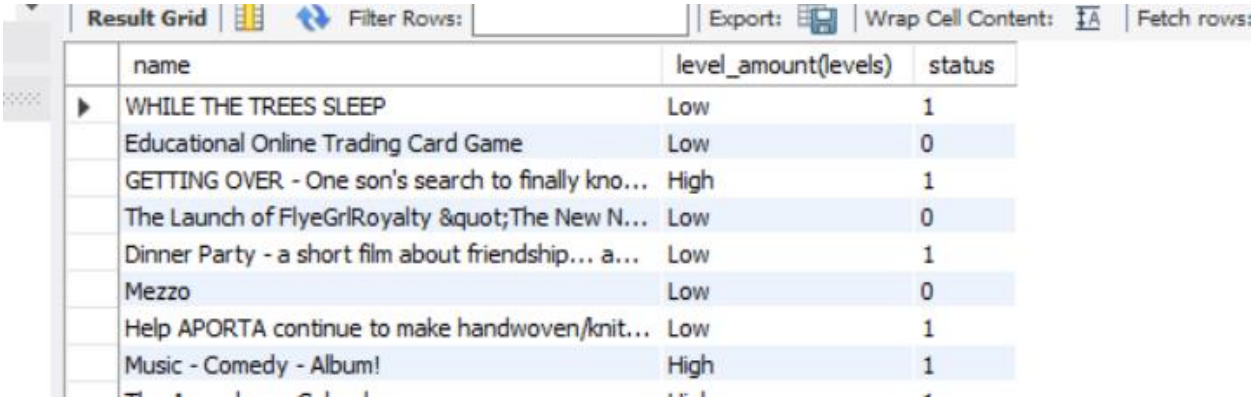



#11

Query 15 with function:

```
select name, goal_status(goal), status from main;
```

Output:

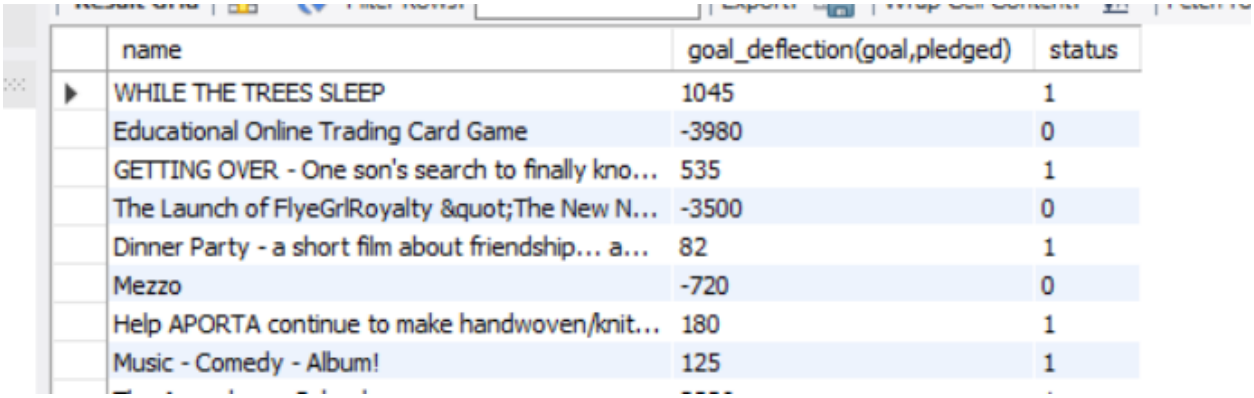


#12

Query 16 with function:

select name, level_amount(levels), status from main;

Output:



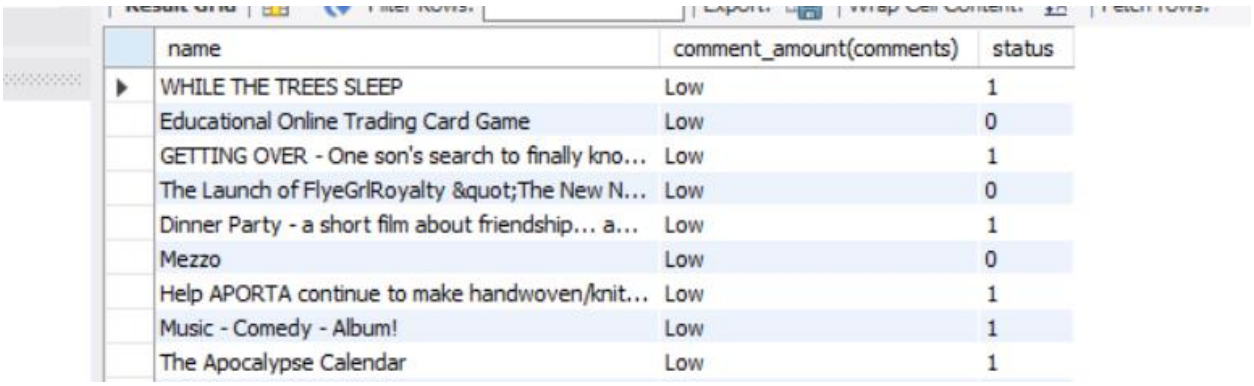
| name | goal_deflection(goal,pledged) | status |
|---|---|---|
| WHILE THE TREES SLEEP | 1045 | 1 |
| Educational Online Trading Card Game | -3980 | 0 |
| GETTING OVER - One son's search to finally kno... | 535 | 1 |
| The Launch of FlyeGrlRoyalty "The New N... | -3500 | 0 |
| Dinner Party - a short film about friendship... a... | 82 | 1 |
| Mezzo | -720 | 0 |
| Help APORTA continue to make handwoven/knit... | 180 | 1 |
| Music - Comedy - Album! | 125 | 1 |

#13

Query 17 with function:

select name, goal_deflection(goal,pledged), status from main;

Output:



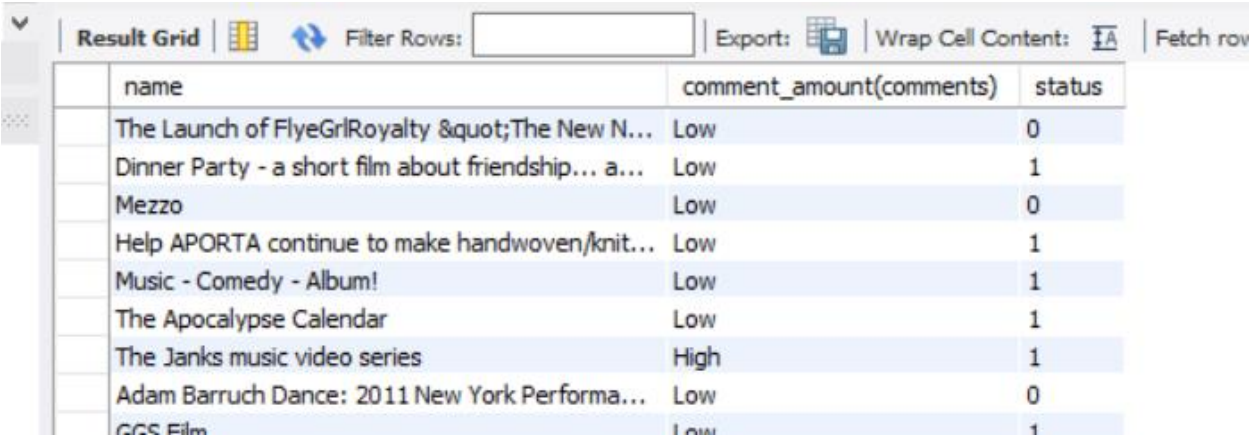
| name | comment_amount(comments) | status |
|---|---|---|
| WHILE THE TREES SLEEP | Low | 1 |
| Educational Online Trading Card Game | Low | 0 |
| GETTING OVER - One son's search to finally kno... | Low | 1 |
| The Launch of FlyeGrlRoyalty "The New N... | Low | 0 |
| Dinner Party - a short film about friendship... a... | Low | 1 |
| Mezzo | Low | 0 |
| Help APORTA continue to make handwoven/knit... | Low | 1 |
| Music - Comedy - Album! | Low | 1 |
| The Apocalypse Calendar | Low | 1 |

#14

Query 18 with function:

select name, comment_amount(comments), status from main;

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rov

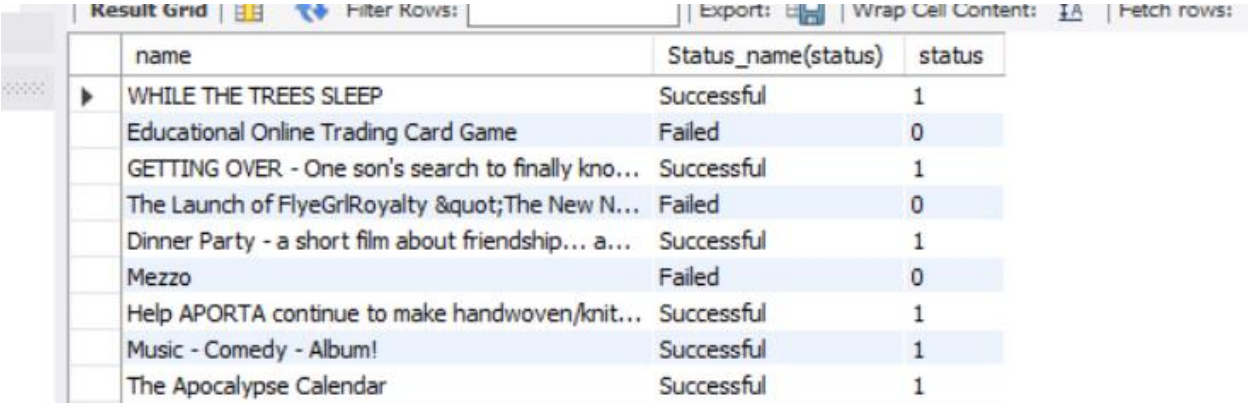
| name | comment_amount(comments) | status |
|---|---|---|
| The Launch of FlyeGrlRoyalty "The New N... | Low | 0 |
| Dinner Party - a short film about friendship... a... | Low | 1 |
| Mezzo | Low | 0 |
| Help APORTA continue to make handwoven/knit... | Low | 1 |
| Music - Comedy - Album! | Low | 1 |
| The Apocalypse Calendar | Low | 1 |
| The Janks music video series | High | 1 |
| Adam Barruch Dance: 2011 New York Performa... | Low | 0 |
| GGS Film | Low | 1 |

#15

Query 19 with function:

select name, Status_name(status), status from main;

Output:

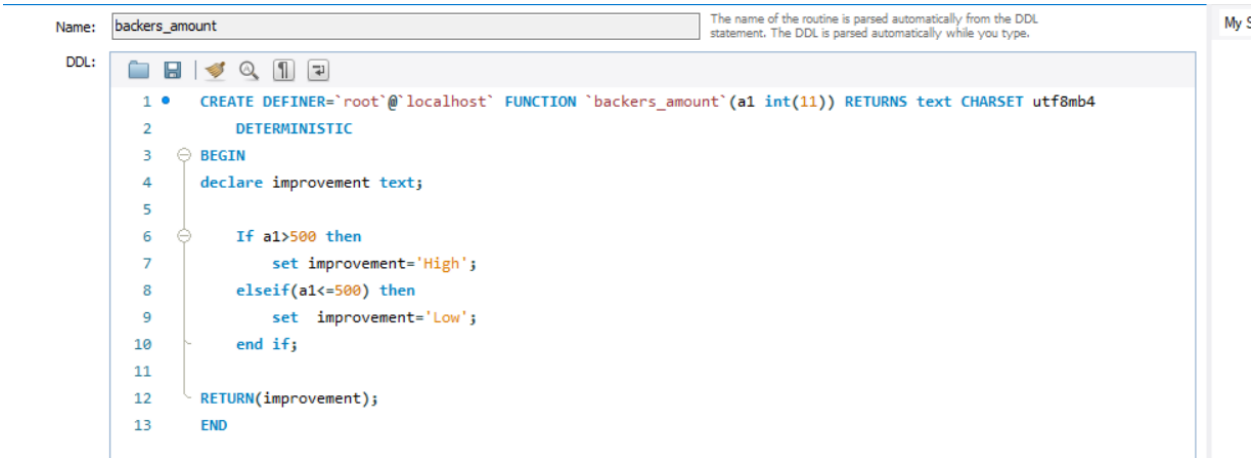


Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

| name | Status_name(status) | status |
|---|---|---|
| WHILE THE TREES SLEEP | Successful | 1 |
| Educational Online Trading Card Game | Failed | 0 |
| GETTING OVER - One son's search to finally kno... | Successful | 1 |
| The Launch of FlyeGrlRoyalty "The New N... | Failed | 0 |
| Dinner Party - a short film about friendship... a... | Successful | 1 |
| Mezzo | Failed | 0 |
| Help APORTA continue to make handwoven/knit... | Successful | 1 |
| Music - Comedy - Album! | Successful | 1 |
| The Apocalypse Calendar | Successful | 1 |

# Functions in SQL

## Function 1:




## Function 2:


## Function 3:

## Function4:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `goal_status`(a1 int(11)) RETURNS text CHARSET utf8mb4
    DETERMINISTIC
BEGIN
declare improvement text;

    If a1>5000 then
        set improvement='High';
    elseif(a1<=5000) then
        set  improvement='Low';
    end if;

RETURN(improvement);
END
```

## Function 5:

DDL:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `level_amount`(a1 int(11)) RETURNS text CHARSET utf8mb4
    DETERMINISTIC
BEGIN
declare improvement text;

    If a1>7 then
        set improvement='High';
    elseif(a1<=7) then
        set  improvement='Low';
    end if;

RETURN(improvement);
END
```

## Function 6:

DDL:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `Status_Name`(a1 int(11)) RETURNS text CHARSET utf8mb4
    DETERMINISTIC
BEGIN
declare improvement text;

    If a1=1 then
        set improvement='Successful';
    elseif(a1=0) then
        set  improvement='Failed';
    end if;

RETURN(improvement);
END
```

# Views in SQL:

## View 1:

DDL:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `project`.`view1` AS
    SELECT
        `project`.`main`.`category` AS `Category`,
        (AVG(`project`.`main`.`status`) * 100) AS `percentage`
    FROM
        `project`.`main`
    GROUP BY `project`.`main`.`category`
```

## View 2:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `project`.`view2` AS
    SELECT
        `project`.`state`.`State` AS `state`,
        SUM(`project`.`main`.`goal`) AS `sum(goal)`
    FROM
        (`project`.`main`
        LEFT JOIN `project`.`state` ON ((`project`.`main`.`state` = `project`.`state`.`State_id`)))
    WHERE
        (`project`.`main`.`status` = 1)
    GROUP BY `project`.`state`.`State`
    ORDER BY SUM(`project`.`main`.`goal`) DESC
```

## View 3:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `project`.`view3` AS
    SELECT
        `project`.`subcategory`.`Subcategory_Name` AS `subcategory_name`,
        SUM(`project`.`main`.`backers`) AS `Trending`
    FROM
        (`project`.`main`
        LEFT JOIN `project`.`subcategory` ON ((`project`.`main`.`subcategory` = `project`.`subcategory`.`Subcateg
    GROUP BY `project`.`main`.`subcategory`
    ORDER BY SUM(`project`.`main`.`backers`) DESC
```

## View 4:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `project`.`view4` AS
    SELECT
        `project`.`main`.`unique_id` AS `unique_id`,
        `project`.`main`.`project id` AS `project id`,
        `project`.`main`.`name` AS `name`,
        `project`.`main`.`url` AS `url`,
        `project`.`main`.`category` AS `category`,
        `project`.`main`.`subcategory` AS `subcategory`,
        `project`.`main`.`status` AS `status`,
        `project`.`main`.`goal` AS `goal`,
        `project`.`main`.`pledged` AS `pledged`,
        `project`.`main`.`funded percentage` AS `funded percentage`,
        `project`.`main`.`backers` AS `backers`,
```

## View 5:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `project`.`view5` AS
    SELECT
        AVG(`project`.`main`.`goal`) AS `Average_Goal_of_Successful_Project`
    FROM
        `project`.`main`
    WHERE
        (`project`.`main`.`status` = 1)
```

## View 6:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `project`.`view6` AS
    SELECT
        `project`.`main`.`unique_id` AS `unique_id`,
        `project`.`main`.`project id` AS `project id`,
        `project`.`main`.`name` AS `name`,
        `project`.`main`.`url` AS `url`,
        `project`.`main`.`category` AS `category`,
        `project`.`main`.`subcategory` AS `subcategory`,
        `project`.`main`.`status` AS `status`,
        `project`.`main`.`goal` AS `goal`,
        `project`.`main`.`pledged` AS `pledged`,
        `project`.`main`.`funded percentage` AS `funded percentage`,
        `project`.`main`.`backers` AS `backers`,
```

**References:**


- **Kickstarter** (https://www.kickstarter.com/)
- **Predicting the success of Kickstarter campaigns (**https://towardsdatascience.com/predicting-the-success-of-kickstarter-campaigns-3f4a976419b9**)**
- **Kaggle (**https://www.kaggle.com/parienza/kickstarter**)**


# License