1. Write a query that returns a list of all the unique values in the 'country' field.

SELECT DISTINCT(country)

FROM ksprojects;

1. How many unique values are there for the main\_category field? What about for the category field?

Unique main\_category Count = 15

Unique category Count = 158

1. Get a list of all the unique combinations of main\_category and category fields, sorted A to Z by main\_category.

SELECT DISTINCT(main\_category), (category)

FROM ksprojects

ORDER BY main\_category ASC, category ASC;

(I added an extra order by because I wanted it to look better lol).

1. How many unique categories are in each main\_category?

SELECT DISTINCT main\_category, COUNT(DISTINCT category)

FROM ksprojects

GROUP BY main\_category

ORDER BY main\_category ASC;

1. Write a query that returns the average number of backers per main\_category, rounded to the nearest whole number and sorted from high to low.

SELECT main\_category, ROUND(AVG(backers),0) AS avg\_num\_backers

FROM ksprojects

GROUP BY main\_category

ORDER BY avg\_num\_backers DESC;

1. Write a query that shows, for each category, how many campaigns were successful and the average difference per project between dollars pledged and the goal.

SELECT category, COUNT(state), AVG(pledged - goal) AS avg\_over\_goal

FROM ksprojects

WHERE state = 'successful'

GROUP BY category;

1. Write a query that shows, for each main category, how many projects had zero backers for that category and the largest goal amount for that category (also for projects with zero backers).

SELECT main\_category, COUNT(backers) AS num\_zero\_backers, MAX(goal) AS max\_goal

FROM ksprojects

WHERE backers = 0

GROUP BY main\_category;

1. For each category, find the average USD per backer, and return only those results for which the average USD per backer is < $50, sorted high to low. Hint: Division by NULL is not possible, so use NULLIF to replace NULLs with 0 in the average calculation.

SELECT category, AVG(usd\_pledged/NULLIF(backers,0)) AS avg\_per\_backer

FROM ksprojects

GROUP BY category

HAVING AVG(usd\_pledged/NULLIF(backers,0)) < 50

ORDER BY avg\_per\_backer DESC;

1. Write a query that shows, for each main\_category, how many successful projects had between 5 and 10 backers.

SELECT main\_category, COUNT(state) as num\_successful

FROM ksprojects

WHERE state = 'successful' AND

backers BETWEEN 5 AND 10

GROUP BY main\_category;

1. Get a total of the amount ‘pledged’ for each type of currency grouped by its respective currency. Sort by ‘pledged’ from high to low.

SELECT currency, SUM(pledged) AS pledged

FROM ksprojects

GROUP BY currency

ORDER BY SUM(pledged) DESC;

1. Excluding Games and Technology records in the main\_category field, return the total of all backers for successful projects by main\_category where the total was more than 100,000. Sort by main\_category from A to Z.

SELECT main\_category, SUM(backers) as total\_backers

FROM ksprojects

WHERE

state = 'successful' AND

main\_category NOT IN( 'Technology', 'Games')

GROUP BY main\_category

HAVING SUM(backers) > 100000

ORDER BY main\_category ASC;