1. Top 5 most expensive projects by country (TESTED - WORKING)

SELECT top 5 agencies.countryName, funding.projectID, funding.agencyAcr, funding.cost

FROM funding LEFT JOIN agencies ON funding.agencyAcr = agencies.agencyAcr

GROUP BY agencies.countryName, funding.projectID, funding.agencyAcr, funding.cost

ORDER BY funding.cost DESC;

2. Top 5 least expensive projects (TESTED -WORKING)

SELECT top 5 agencies.countryName, funding.projectID, funding.agencyAcr, funding.cost

FROM funding LEFT JOIN agencies ON funding.agencyAcr = agencies.agencyAcr

GROUP BY agencies.countryName, funding.projectID, funding.agencyAcr, funding.cost

HAVING countryName is NOT NULL

ORDER BY funding.cost ASC;

3. Search for an astronaut given a string- search first and last name, report first name, last name and their country name – TESTED WORKING

SELECT astronaut.firstname, astronaut.lastname,country.countryName

FROM astronaut

JOIN nationality ON astronaut.personID= nationality.personID

JOIN country ON nationality.countryName=country.countryName

WHERE astronaut.firstname LIKE ‘a%’

AND astronaut.lastname LIKE ‘%a’

4. The launch month with the highest successful space missions? TESTED WORKING

WITH successMissions AS (

SELECT projects.projectID, projects.missionStatus

FROM projects

WHERE projects.missionStatus = 'success'

)

SELECT TOP 1 projects.launchMonth , COUNT(projects.launchMonth) AS numSuccessfulMissions

FROM projects

WHERE projects.projectID IN (

SELECT projectID FROM successMissions

)

GROUP BY projects.launchMonth

ORDER BY numSuccessfulMissions DESC

--reutrns launchMonth, numSuccessfulMissions

5. Based on the average funding amount, report the number of projects that failed given that they are below the average, and the number of projects that succeeded given that they are below average TESTED, WORKING

WITH avgCost AS( SELECT COUNT(funding.projectID) AS totalFundedProjects, AVG(funding.cost) AS averageCostFROM funding ), belowAvg AS( SELECT funding.projectID, funding.cost FROM funding WHERE funding.cost < (SELECT averageCost FROM avgCost) ), aboveAvg AS(SELECT funding.projectID, funding.cost FROM funding WHERE funding.cost > (SELECT averageCost from avgCost) ), successMissions AS(SELECT projects.projectID, funding.cost FROM projects LEFT JOIN funding ON projects.projectID = funding.projectID WHERE projects.missionStatus = 'success' ), numSuccessBelow AS( SELECT averageCost, totalFundedProjects, COUNT(funding.projectID) as numSuccessBelow FROM funding CROSS JOIN avgCost WHERE funding.projectID IN (SELECT projectID FROM belowAvg) AND funding.projectID IN (SELECT projectID FROM successMissions) GROUP BY averageCost , totalFundedProjects ), numFailedBelow AS( SELECT averageCost, COUNT(funding.projectID) as numFailedBelow FROM funding CROSS JOIN avgCost WHERE funding.projectID IN (SELECT projectID FROM belowAvg) AND funding.projectID NOT IN (SELECT projectID FROM successMissions) GROUP BY averageCost ),

numSuccessAbove AS(SELECT averageCost, totalFundedProjects, COUNT(funding.projectID) as numSuccessAbove FROM funding CROSS JOIN avgCost WHERE funding.projectID IN (SELECT projectID FROM aboveAvg)

AND funding.projectID IN (SELECT projectID FROM successMissions)

GROUP BY averageCost , totalFundedProjects

),

numFailedAbove AS(

SELECT averageCost, COUNT(funding.projectID) as numFailedAbove

FROM funding CROSS JOIN avgCost

WHERE funding.projectID IN (SELECT projectID FROM aboveAvg)

AND funding.projectID NOT IN (SELECT projectID FROM successMissions)

GROUP BY averageCost )

SELECT numSuccessBelow.averageCost, numSuccessBelow, numFailedBelow, numSuccessAbove, numFailedAbove, numSuccessBelow.totalFundedProjects

FROM numSuccessBelow LEFT JOIN numFailedBelow ON

numSuccessBelow.averageCost = numFailedBelow.averageCost

LEFT JOIN numSuccessAbove ON numSuccessAbove.averageCost = numFailedBelow.averageCost

LEFT JOIN numFailedAbove ON numSuccessAbove.averageCost = numFailedAbove.averageCost

WITH avgCost AS( SELECT COUNT(funding.projectID) AS totalFundedProjects, AVG(funding.cost) AS averageCost FROM funding ), belowAvg AS(SELECT funding.projectID, funding.cost FROM funding WHERE funding.cost < (SELECT averageCost FROM avgCost) ),

aboveAvg AS(

SELECT funding.projectID, funding.cost FROM funding

WHERE funding.cost > (SELECT averageCost from avgCost)

),

successMissions AS(

SELECT projects.projectID, funding.cost FROM projects LEFT JOIN funding ON projects.projectID = funding.projectID

WHERE projects.missionStatus = 'success'

),

numSuccessBelow AS(

SELECT averageCost, totalFundedProjects, COUNT(funding.projectID) as numSuccessBelow

FROM funding CROSS JOIN avgCost

WHERE funding.projectID IN (SELECT projectID FROM belowAvg)

AND funding.projectID IN (SELECT projectID FROM successMissions)

GROUP BY averageCost , totalFundedProjects

),

numFailedBelow AS(

SELECT averageCost, COUNT(funding.projectID) as numFailedBelow

FROM funding CROSS JOIN avgCost

WHERE funding.projectID IN (SELECT projectID FROM belowAvg)

AND funding.projectID NOT IN (SELECT projectID FROM successMissions)

GROUP BY averageCost

),

numSuccessAbove AS(

SELECT averageCost, totalFundedProjects, COUNT(funding.projectID) as numSuccessAbove

FROM funding CROSS JOIN avgCost

WHERE funding.projectID IN (SELECT projectID FROM aboveAvg)

AND funding.projectID IN (SELECT projectID FROM successMissions)

GROUP BY averageCost , totalFundedProjects

),

numFailedAbove AS(

SELECT averageCost, COUNT(funding.projectID) as numFailedAbove

FROM funding CROSS JOIN avgCost

WHERE funding.projectID IN (SELECT projectID FROM aboveAvg)

AND funding.projectID NOT IN (SELECT projectID FROM successMissions)

GROUP BY averageCost )

SELECT numSuccessBelow.averageCost, numSuccessBelow, numFailedBelow, numSuccessAbove, numFailedAbove, numSuccessBelow.totalFundedProjects

FROM numSuccessBelow LEFT JOIN numFailedBelow ON

numSuccessBelow.averageCost = numFailedBelow.averageCost

LEFT JOIN numSuccessAbove ON numSuccessAbove.averageCost = numFailedBelow.averageCost

LEFT JOIN numFailedAbove ON numSuccessAbove.averageCost = numFailedAbove.averageCost

------------------

6. All astronaut who has been in a failed mission -- not doable (DELETe?)

select distinct astronaut.firstname ,astronaut.lastname

from astronaut

join nationality on astronaut.personID = nationality.personID

join country on nationality.countryName = country.countryName

join agencies on nationality.countryName= agencies.countryName

join funding on agencies. agencyAcr = funding. agencyAcr

join projects on funding.projectID=projects.projectID

where projects.missionStatus= 'failure';

6. Return all manufacturers founded in the 21st century that have built more than 5 spacecrafts- TESTED, WORKING

SELECT manufacturers.manufacturerName,COUNT(manufacturers.manufacturerName)AS numSpacecrafts FROM builtBy join manufacturers on builtBy.manufacturerName = manufacturers.manufacturerName where yearFound >2000 GROUP BY manufacturers.manufacturerName HAVING(COUNT(manufacturers.manufacturerName) > 5)

7. All the missions between [this year] and [this year] from [this country]-- TESTED, WORKING

WITH allMissFromCountry AS(SELECT projects.projectID FROM projects WHERE projects.countryName = 'United States') SELECT \* FROM projects WHERE projects.projectID IN (SELECT projectID FROM allMissFromCountry) AND projects.flightYear BETWEEN 2015 AND 2020

--replace ‘United States’, 2015 and 2020 with ?

8. List of agencies that have fulfilled at least one activity TESTED WORKING

SELECT \* from agencies WHERE evActivity = 1 OR spaceRD = 1 OR spaceStation = 1 OR humanFlight = 1 OR multipleSAt = 1 OR extraProbe = 1

9. Top 3 astronauts in each country having the most fly time (window functions)

TESTED WORKING

with mostFlyTime as( select nationality.countryName, astronaut.firstname, astronaut.lastname, rank() over (partition by nationality.countryName order by totalFlightTime desc) as Rank from astronaut join nationality on astronaut.personID = nationality.personID ) select \* from mostFlyTime as x where x.Rank < 4

10. Return all satellite names from the launchVehicle which launched the most satellites

-- could possibly be a much simpler query TESTED WORKING

WITH numLaunched AS( SELECT launchVehicle, COUNT(launchVehicle) AS numLaunched FROM satellite GROUP BY launchVehicle ),

mostLaunched AS( SELECT launchVehicle FROM numLaunched WHERE numLaunched =(SELECT MAX(numLaunched) AS mostLaunched FROM numLaunched) ) SELECT satellite.spacecraftName AS satellites FROM satellite WHERE satellite.launchVehicle = (SELECT launchVehicle FROM mostLaunched)

Will also add some basic ones

12. search project by projectID – TESTED, WORKING

select \*

from projects

where projects.projectID=5

-- replace 5 with ?

11.search spacecraft, funding, by projectID – TESTED WORKING

SELECT builtBy.spacecraftName, funding.cost FROM projects LEFT JOIN funding ON projects.projectID = funding.projectID LEFT JOIN builtBy ON funding.agencyAcr = builtBy.manufacturerName WHERE projects.projectID = ? AND builtBy.spacecraftName IS NOT NULL AND funding.cost IS NOT NULL

SELECT spacecraftUsed.spacecraftName, funding.cost FROM projects LEFT JOIN funding ON projects.projectID = funding.projectID LEFT JOIN spacecraftUsed ON spacecraftUsed.projectID = projects.projectID WHERE projects.projectID = ?

--will return nothing if spacecraftname or cost is null (or both), tested on 1 and 14

-search astronauts by projectID – not doable

String sql = "WITH avgCost AS(SELECT COUNT(funding.projectID) AS totalFundedProjects, AVG(funding.cost) AS averageCost FROM funding), "+

"belowAvg AS( "+

"SELECT funding.projectID, funding.cost FROM funding "+

"WHERE funding.cost < (SELECT averageCost FROM avgCost)), "+

"aboveAvg AS( "+

"SELECT funding.projectID, funding.cost FROM funding "+

"WHERE funding.cost > (SELECT averageCost from avgCost) "+

"), "+

"successMissions AS( "+

"SELECT projects.projectID, funding.cost FROM projects LEFT JOIN funding ON projects.projectID = funding.projectID "+

"WHERE projects.missionStatus = 'success' "+

"), "+

"numSuccessBelow AS( "+

"SELECT averageCost, totalFundedProjects, COUNT(funding.projectID) as numSuccessBelow "+

"FROM funding CROSS JOIN avgCost"+

"WHERE funding.projectID IN (SELECT projectID FROM belowAvg) "+

"AND funding.projectID IN (SELECT projectID FROM successMissions) "+

"GROUP BY averageCost , totalFundedProjects "+

"), "+

"numFailedBelow AS( "+

"SELECT averageCost, COUNT(funding.projectID) as numFailedBelow "+

"FROM funding CROSS JOIN avgCost "+

"WHERE funding.projectID IN (SELECT projectID FROM belowAvg) "+

"AND funding.projectID NOT IN (SELECT projectID FROM successMissions) "+

"GROUP BY averageCost ), "+

"numSuccessAbove AS( "+

"SELECT averageCost, totalFundedProjects, COUNT(funding.projectID) as numSuccessAbove "+

"FROM funding CROSS JOIN avgCost "+

"WHERE funding.projectID IN (SELECT projectID FROM aboveAvg) "+

"AND funding.projectID IN (SELECT projectID FROM successMissions) "+

"GROUP BY averageCost , totalFundedProjects "+

"), "+

"numFailedAbove AS( "+

"SELECT averageCost, COUNT(funding.projectID) as numFailedAbove "+

"FROM funding CROSS JOIN avgCost "+

"WHERE funding.projectID IN (SELECT projectID FROM aboveAvg) "+

"AND funding.projectID NOT IN (SELECT projectID FROM successMissions) "+

"GROUP BY averageCost "+

") " +

"SELECT numSuccessBelow.averageCost, numSuccessBelow, numFailedBelow, numSuccessAbove, numFailedAbove, numSuccessBelow.totalFundedProjects "+

"FROM numSuccessBelow LEFT JOIN numFailedBelow ON "+

"numSuccessBelow.averageCost = numFailedBelow.averageCost "+

"LEFT JOIN numSuccessAbove ON numSuccessAbove.averageCost = numFailedBelow.averageCost "+

"LEFT JOIN numFailedAbove ON numSuccessAbove.averageCost = numFailedAbove.averageCost";

10. Return all satellite names from the launchVehicle which launched the most satellites

-- could possibly be a much simpler query TESTED WORKING

String sql = “WITH numLaunched AS( SELECT launchVehicle, COUNT(launchVehicle) AS “ + “numLaunched FROM satellite GROUP BY launchVehicle ), “ +

“mostLaunched AS( SELECT launchVehicle FROM numLaunched WHERE numLaunched “ + “=(SELECT MAX(numLaunched) AS mostLaunched FROM numLaunched) ) SELECT “ + “satellite.spacecraftName AS satellites FROM satellite WHERE “ + “satellite.launchVehicle = (SELECT launchVehicle FROM mostLaunched)”