National University of Computer and Emerging Sciences

Lahore Campus

Database System

Date:

Instructor(s)

Mr. Muhammad Naveed

Assignment 2

Course code: CS2005

Total Time:

Total marks: 100

Questions: 03

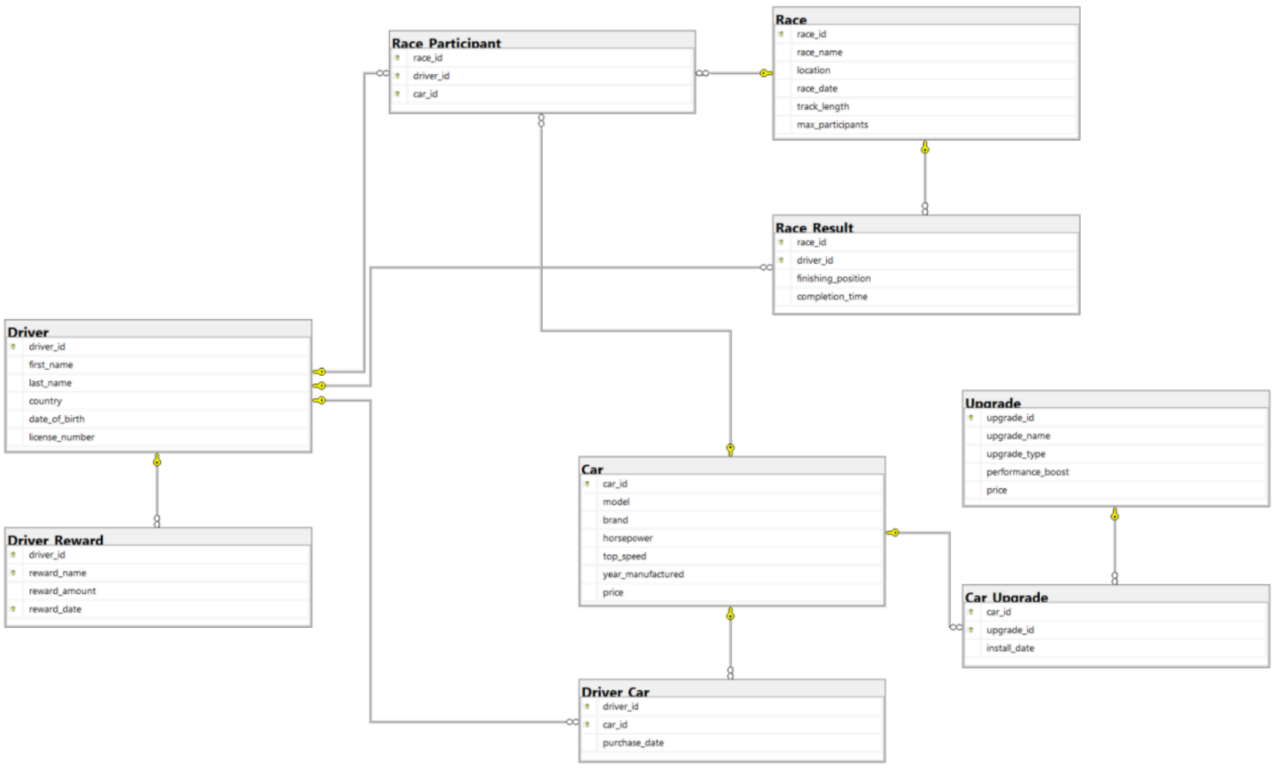
_____ _____ _____

Roll No Section Student Signature

Attempt all the questions.

CLO5 Author queries using relational algebra and SQL

Question no 1: **30 marks**



In the Fast & Furious Racing League, elite racers from around the world compete in high-speed underground races. Each driver owns multiple cars, which can be customized with upgrades. Races happen at different locations, and each race has multiple participants. The winners earn rewards and ranking points. To help the DBA manage functionalities of Fast & Furious, write the following queries using **JOINS and AGGREGATE FUNCTION also, create tables and insert data in them from the provided file only.**

1. List all drivers and their cars.
2. Show all race names and the number of participants
3. Display the fastest driver in each race
4. Find the total prize money won by each driver.
5. List all upgrades installed on each car.
6. Find drivers who have participated in at least 2 races.
7. Display the average completion time for each race.
8. Find the driver who owns the most expensive car.
9. Show the ranking of drivers based on total prize money won.
10. List all drivers who own at least two different car brands.
11. Find drivers who have participated in every race.
12. Find the top 3 drivers with the fastest average race completion time.
13. Display drivers who own cars with at least 2 different upgrades.

QUESTION NO 2 **30 marks**

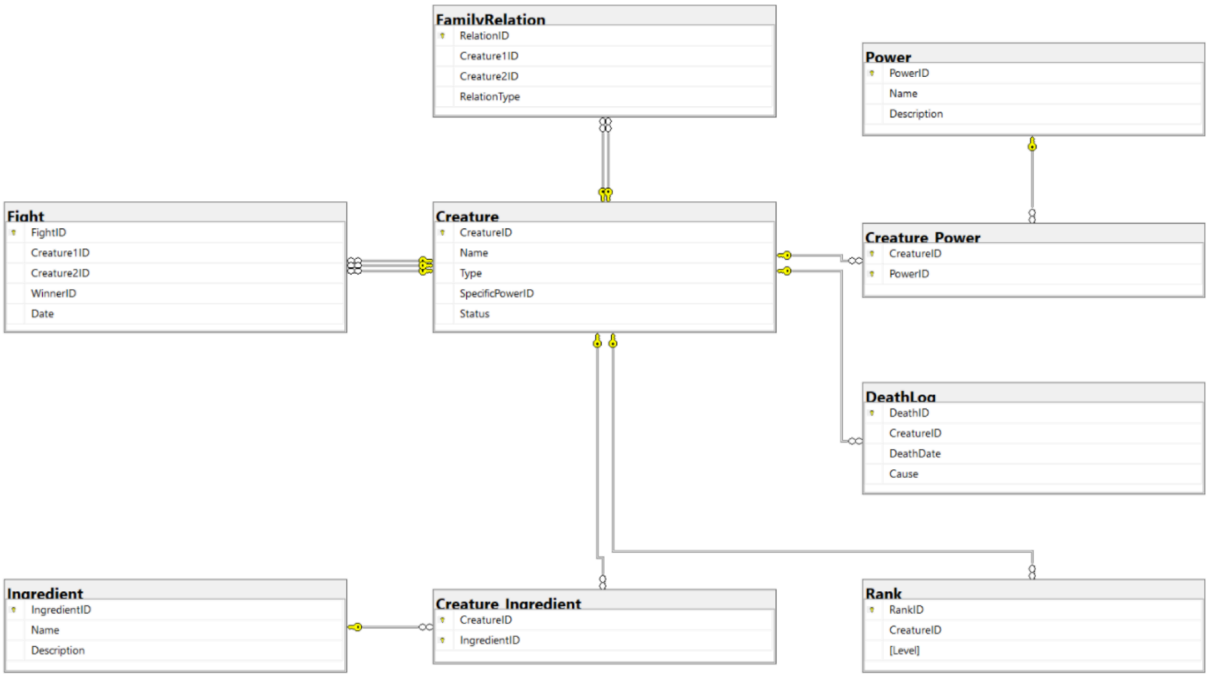
PowerPuffDB: The Battle Records

In Townsville, battles between heroes and villains shape history. PowerPuffDB is the sacred archive tracking every hero, villain, power, and fight.

The Rules of Power

- Bloodlines Cannot Be Broken – Family ties must be preserved, preventing automatic deletions.
- Fights Must Be Logged ⚔️ – Every battle is recorded, but victors cannot vanish without a trace.
- Balance is Key ⚖️ – Powers and ranks follow strict rules, ensuring fairness in the chaos.

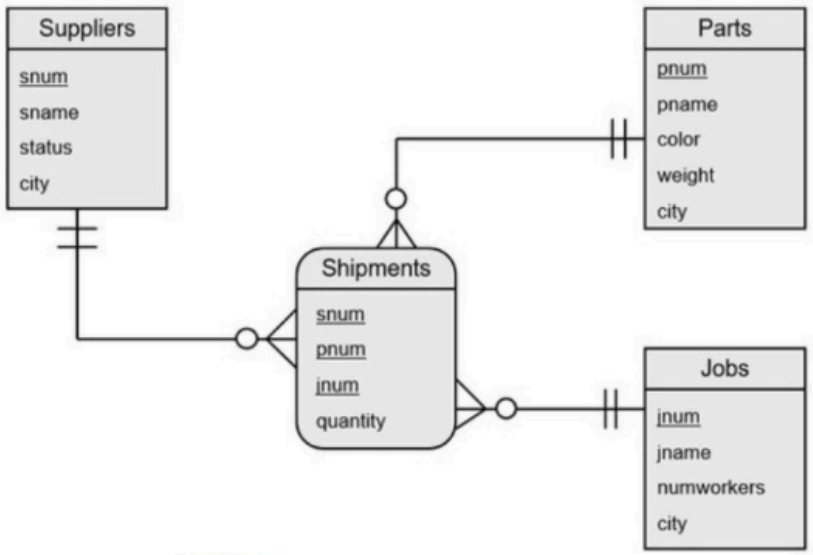
As the Guardian of PowerPuffDB, your mission is to uphold order. Code wisely, history depends on it. Help writing the following queries using **JOINS and AGGREGATE FUNCTION. Also, create tables and insert data in them from the provided file only.**

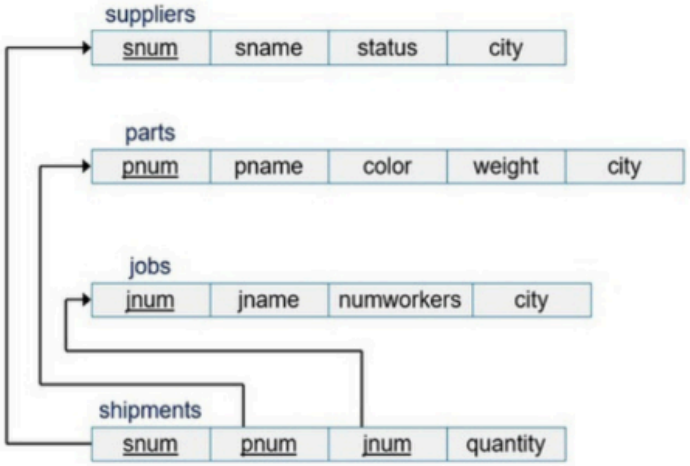


Find the name of the strongest creature (highest rank).
Get the most frequently used ingredient.
Retrieve the creature that fought the most battles.
Find creatures with only one power.
Retrieve creatures that have never lost a fight.
Get the youngest sibling (highest CreatureID).
Find the creature with the most powers.
Get creatures that fought at least twice.
Find creatures that died and had a rank.
Find creatures that used ‘Chemical X’ in their creation.
Find the creature that has fought against the most different opponents.
Find pairs of creatures that are siblings and have fought each other at least once.
List creatures with their powers, ranks, and number of battles, ordered by a combined “strength score” (e.g., number of powers × rank level × battles).

Question no 3 **40 marks**

Read the following database scheme carefully. Make an understanding of table with primary and foreign key relations and implement the tables.





supplier

S#	SName	Status	City
S1	SMITH	20	LONDON
S2	JONES	10	PARIS
S3	BLAKE	30	PARIS
S4	CLARK	20	LONDON
S5	ADAMS	30	ATHENS

part

P#	PName	Color	City
P1	NUT	RED	LONDON
P2	BOLT	GREEN	PARIS
P3	SCREW	BLUE	ROME
P4	SCREW	RED	LONDON
P5	CAM	BLUE	PARIS
P6	COG	RED	LONDON

jobs

J#	JName	numworkers	City
J1	SORTER	20	PARIS
J2	PUNCH	10	ROME
J3	READER	30	ATHENS
J4	CONSOLE	20	ATHENS
J5	COLLATOR	30	LONDON
J6	TERMINAL	20	OSLO
J7	TAPE	10	LONDON

shipments

S#	P#	J#	QTY
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800
S2	P5	J2	100
S3	P3	J1	200
S3	P4	J2	500
S4	P6	J3	300
S4	P6	J7	300
S5	P2	J2	200
S5	P2	J4	100
S5	P5	J5	500
S5	P5	J7	100
S5	P6	J2	200
S5	P1	J4	100
S5	P3	J4	200
S5	P4	J4	800
S5	P5	J4	400
S5	P6	J4	500

Sample table data is attached for your reference.

1. List the part number for every part that is shipped by more than one supplier.

2. Find the average weight of all parts.

3. For each part list the part number and the total quantity in which that part is shipped and order the results in descending order of the total quantity shipped. Name the total quantity shipped in the result as total Shipped.

4. List only the names of those suppliers who ship a part that weighs more than 200.

5. List the names of those cities in which both a supplier and a job are located.

6. List the names of those jobs that receive a shipment from supplier number S1.

7. List the names of those parts that are not shipped to any job.

8. List the names of those suppliers who ship part number P2 to any job.

9. List the names of those suppliers who ship part at least one red part to any job.

10. List the part number for every part that is shipped more than once (the part must be shipped more than one time).