Q1) Consider the following Employee table.

emp_id	emp_name	job_name	manager_id	hire_date	salary
68319	KAYLING	PRESIDENT		1991-11-18	6000.00
66928	BLAZE	MANAGER	68319	1991-05-01	2750.00
67832	CLARE	MANAGER	68319	1991-06-09	2550.00
65646	JONAS	MANAGER	68319	1991-04-02	2957.00
67858	SCARLET	ANALYST	65646	1997-04-19	3100.00

Write SQL commands to do the following

Q1) List the average salary of each job title, sorted in descending order of average salary:

```
SELECT job_name, AVG(salary) AS avg_salary
FROM Employee
GROUP BY job_name
ORDER BY avg_salary DESC;
```

Output:

job_name	avg_salary
PRESIDENT	6000
ANALYST	3100
MANAGER	2752.333
SALESMAN	1500
CLERK	1137.5

Q2) Find the job title and the number of employees whose salary is greater than the average salary of their respective job titles, sorted by job title:

```
SELECT job_name, COUNT(*) AS num_employees
FROM Employee
GROUP BY job_name
HAVING AVG(salary) < (SELECT AVG(salary) FROM Employee WHERE job_name =
Employee.job_name)
ORDER BY job_name;
```

Output:

job_name	num_employees
CLERK	4
SALESMAN	4

Q3) List the manager_id and the number of employees managed by each manager who manages more than one employee, sorted by manager_id:

```
SELECT manager_id, COUNT(*) AS num_managed_employees
FROM Employee
GROUP BY manager_id
HAVING COUNT(*) > 1
ORDER BY manager_id;
```

Output:

manager_id	num_managed_employees
65646	2
66928	5
68319	3

Q2) Consider the following Worker table.

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
001	Monika	Arora	100000	2021-02-20 09:00:00	HR
002	Niharika	Verma	80000	2021-06-11 09:00:00	Admin
003	Vishal	Singhal	300000	2021-02-20 09:00:00	HR
004	Amitabh	Singh	500000	2021-02-20 09:00:00	Admin
005	Vivek	Bhati	500000	2021-06-11 09:00:00	Admin

Write SQL commands to do the following

Q1) List the department and the total salary for each department, sorted by total salary in descending order:

```
SELECT Department, SUM(Salary) AS total_salary
FROM Worker
GROUP BY Department
ORDER BY total_salary DESC;
```

Output:

Department	total_salary
Admin	1170000
HR	400000
Account	275000

Q2) Find the average salary for each department and display only those departments where the average salary is above $100,000, sorted by average salary in descending order:

```
SELECT Department, AVG(Salary) AS avg_salary
FROM Worker
GROUP BY Department
HAVING AVG(Salary) > 100000
ORDER BY avg_salary DESC;
```

Output:

Department	avg_salary
Admin	292500
HR	200000
Account	137500

Q3) List the number of workers in each department who joined after January 1, 2021, sorted by department name:

```
SELECT Department, COUNT(*) AS num_workers
FROM Worker
WHERE JOINING_DATE > '2021-01-01'
GROUP BY Department
ORDER BY Department;
```

Output:

Department	num_workers
Account	2
Admin	4
HR	2

Q3) Consider the following table.

PatientID	Name	DateOfBirth	Gender	admit_date	ward_no	City
101	John Smith	1985-07-15	Male	2024-02-10	101	New York
102	Emily Johnson	1990-04-25	Female	2024-02-12	102	Los Angeles
103	Michael Brown	1978-12-10	Male	2024-02-15	103	Chicago
104	Emma Davis	1989-09-08	Female	2024-02-20	104	Houston
105	Olivia Wilson	1973-03-30	Female	2024-02-22	105	Miami

Write SQL commands to do the following

Q1) List the number of patients admitted to each ward, sorted by ward number:

```
SELECT ward_no, COUNT(*) AS num_patients
FROM Patients
GROUP BY ward_no
ORDER BY ward_no;
```

Output:

ward_no	num_patients
101	1
102	1
103	1
104	1
105	1

Q2) Find the average age of patients admitted to each ward, and display only those wards where the average age is below 40, sorted by average age in descending order:

```
SELECT ward_no, AVG(DATEDIFF(CURRENT_DATE(), DateOfBirth) / 365) AS avg_age
FROM Patients
GROUP BY ward_no
HAVING avg_age < 40
ORDER BY avg_age DESC;
```

Output:

ward_no	avg_age
101	38.6932
104	34.5397
102	33.9123

Q3) List the number of male and female patients admitted to each ward, sorted by ward number and gender:

```
SELECT ward_no, Gender, COUNT(*) AS num_patients
FROM Patients
GROUP BY ward_no, Gender
ORDER BY ward_no, Gender;
```

Output:

ward_no	Gender	num_patients
101	Male	1
102	Female	1
103	Male	1
104	Female	1
105	Female	1

Q4) Consider the following Books table.

book_id	title	author	publication_year	language	available_copies	total_copies
101	The Great Gatsby	F. Scott Fitzgerald	1925	English	10	15
102	To Kill a Mockingbird	Harper Lee	1960	English	5	10
103	1984	George Orwell	1949	English	8	8
104	Pride and Prejudice	Jane Austen	1813	English	12	15
105	The Catcher in the Rye	J.D. Salinger	1951	English	3	5

Write SQL commands to do the following

Q1) List the number of books published in each year, sorted by publication year in ascending order:

```
SELECT publication_year, COUNT(*) AS num_books_published
FROM book
GROUP BY publication_year
ORDER BY publication_year ASC;
```

Output:

publication_year	num_books_published
1813	1
1925	1
1949	1
1951	1
1960	1

Q2) List the authors along with the total number of books they have written, and display only those authors who have written books with more than 10 total copies available, sorted by the number of books in descending order:

```
SELECT author, COUNT(*) AS num_books
FROM book
WHERE total_copies > 10
GROUP BY author
HAVING COUNT(*) > 1
ORDER BY num_books DESC;
```

Output:

author	num_books
F. Scott Fitzgerald	1
Jane Austen	1

Q3) List the titles of books along with the average number of available copies per book, and display only those books where the average number of available copies is less than 5, sorted by average available copies in descending order:

```
SELECT title, AVG(available_copies) AS avg_available_copies
FROM book
GROUP BY title
HAVING AVG(available_copies) < 5
ORDER BY avg_available_copies DESC;
```

Output:

title	avg_available_copies
The Catcher in the Rye	3

Q5) Consider the following Courses table.

CourseID	Title	Instructor	Category	Price	Duration	EnrollmentCount
101	Introduction to Python	John Smith	Programming	33.22	6 weeks	500
102	English Grammar Essentials	Emily Johnson	Language	50.00	8 weeks	750
103	Mathematics for Beginners	Michael Brown	Mathematics	23.32	10 weeks	600
104	History of Ancient Civilizations	Emma Davis	History	100.00	12 weeks	450
105	Web Development Fundamentals	Olivia Wilson	Programming	99.99	8 weeks	800

Q1) List the number of courses in each category, sorted by category name:

```
SELECT Category, COUNT(*) AS num_courses
FROM Courses
GROUP BY Category
ORDER BY Category;
```

Output:

Category	num_courses
History	1
Language	1
Mathematics	1
Programming	2

Q2) Find the categories where the average duration of courses is less than 8 weeks, sorted by category name:

```
SELECT Category, AVG(Duration) AS avg_duration
FROM Courses
GROUP BY Category
HAVING AVG(Duration) < 8
ORDER BY Category;
```

Output:

Category	avg_duration
Programming	7

Q3) List the instructors along with the total number of courses they teach, and display only those instructors who teach courses with more than 500 enrollments, sorted by the number of courses in descending order:

```
SELECT Instructor, COUNT(*) AS num_courses
FROM Courses
GROUP BY Instructor
HAVING SUM(EnrollmentCount) > 500
ORDER BY num_courses DESC;
```

Output:

Instructor	num_courses
Emily Johnson	1
Michael Brown	1
Olivia Wilson	1