Name	ID	STUDENT SIGN

Class Test 01

Hogwarts is a school of witchcraft and wizardry. To ensure proper management of their data the renowned school has decided to maintain a database system. Out of many bidders your company was hired to accomplish the task. Your job is to create a relational database for Hogwarts from the requirements specified below:

RDBMS- Oracle 10g

Language-SQL

Log in as User System and create a ***user*** Dumbledore who has ***password*** Phoenix. Dumbledore is granted ***unlimited tablespace***. He is also granted the permission to ***create*** tables. After logging in with his username and password Dumbledore creates ***two tables*** i.e. Student and House. ***Student*** table has five columns containing information about students ***Identification Number, Name, CGPA, Blood Status and House Number***. ***House*** table has three columns containing information about ***House Number, House Name and House Points***. Here S_Id, H_Id are the ***primary key columns*** of Student and House table respectively. Student table also has a ***foreign key*** column H_No. Constraint should be applied in such a way that CGPA cannot be greater than 4.00 and House name cannot be NULL. The two tables along with their inserted data are given below:

Table: Student

S_Id	S_Name	S_CGPA	S_BloodStatus	H_No
2	Harry	3.45	Halfblood	11
7	Ron	3.01	Pureblood	11
12	Hannah		Pureblood	22
17	Cedric	3.78	Pureblood	22
22	Cho	3.55	Muggleborn	33
27	Luna	2.89		33
32	Draco	3.88	Pureblood	44
37	Goyle	2.10	Pureblood	44

Table: House

H_Id	H_Name	H_Points
11	Gryffindor	892
22	Hufflepuf	785
33	Ravenclaw	789
44	Slytherin	850

After creating the tables and inserting data based on provided requirements write Queries (Write down the question and also the answer. Give screenshot of the result of the query.You can add more Answer Box if required) according to the following specification:

-using **ARITHMETIC** operator

-using **CONCATENATION** operator

-using **COLUMN ALIAS**

-using **LIKE** operator

-using **IS NULL** operator

-using **ORDER BY** clause

-using **SUBSTR** function

-using **NVL** function

-using **MAX** function

-using **SUM** function

-using **GROUP BY** clause

-using **HAVING** clause