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) 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

```
#Creating Student Table

Create table Student(S_Id number(4),S_Name varchar2(20),S_CGPA Float,S_BloodStatus varchar2(20),H_No number(4),

CONSTRAINT pk1 PRIMARY KEY (S_Id));

#Inserting data into Student table

Insert INTO Student(S_Id,S_Name,S_CGPA,S_BloodStatus,H_No) values ('2','Harry','3.45','Halfblood','11');

Insert INTO Student(S_Id,S_Name,S_CGPA,S_BloodStatus,H_No) values ('7','Ron','3.01','Pureblood','11');

Insert INTO Student(S_Id,S_Name,S_CGPA,S_BloodStatus,H_No) values ('12','Hannah',Null,'Pureblood','22');

Insert INTO Student(S_Id,S_Name,S_CGPA,S_BloodStatus,H_No) values ('17','Cedric','3.78','Pureblood','22');

Insert INTO Student(S_Id,S_Name,S_CGPA,S_BloodStatus,H_No) values ('22','Cho','3.55','Muggleborn','33');

Insert INTO Student(S_Id,S_Name,S_CGPA,S_BloodStatus,H_No) values ('27','Luna','2.89',Null,'11');

Insert INTO Student(S_Id,S_Name,S_CGPA,S_BloodStatus,H_No) values ('32','Draco','3.88','Pureblood','44');

Insert INTO Student(S_Id,S_Name,S_CGPA,S_BloodStatus,H_No) values ('37','Goyle','2.10','Pureblood','44');

#Creating Student Table

Create table House(H_Id number(4),H_Name varchar2(20),H_Points number(4),CONSTRAINT pk2 PRIMARY KEY (H_Id));

#Inserting data into Student table

Insert INTO House(H_Id,H_Name,H_Points) values ('11','Gryffindor','892');

Insert INTO House(H_Id,H_Name,H_Points) values ('22','Hufflepuf','785');

Insert INTO House(H_Id,H_Name,H_Points) values ('33','Ravenclaw','789');

Insert INTO House(H_Id,H_Name,H_Points) values ('44','Slytherin','850');
```

1)

Increase The CGPA Of All Students By 0.05

=

```
SELECT S_Name, S_CGPA, S_CGPA+0.005

FROM Student;
```

2)

Display The Student Name And Their CGPA Under 1 Column

=

```
SELECT  S_Name || S_CGPA AS "CGPA"

FROM  Student;
```

3)

Display S_Name as Name and S_CGPA as CGPA

=

```
SELECT S_Name AS name, S_CGPA CGPA

FROM   Student;
```

4)

Display The Student Name, Those Who Have A C At The First Of Their Name

=

```
SELECT        S_Name

FROM          Student

WHERE         S_Name LIKE 'C%';
```

5)

Display The Student Name Who Has No Data Of Their CGPA

=

```
SELECT  S_Name , S_CGPA

FROM    Student

WHERE   S_CGPA IS NULL;
```

6)

Reorder To Ascending Order The House Name By Their H_Points

=

```
SELECT  H_Id ,H_Name, H_Points

FROM    House

ORDER BY H_Points;
```

7)

Use Substr Function And Manupulate The Characters Of Student Name

=

```
SELECT S_Name , CONCAT (S_Name , S_BloodStatus), LENGTH(S_Name ),

INSTR(S_Name , 'o')

FROM   Student
```

```
WHERE SUBSTR(S_BloodStatus,1,5) = 'BloodStatus';
```

8)

Calculate The Housepoint By Multiplying With 10 And Adding Their Corresponding H_Id, Then Display The H_Name, H_Points,H_Id

=

```
SELECT H_Name, H_Points, H_Id , (H_Points*10)+NVL(H_Id ,0)

FROM   House;
```

9)

Display The Avg, Min, Max, And Sum Of H_Points, Those Whose House Points Start With The Numeric Character 7

=

```
SELECT         AVG(H_Points), MAX(H_Points),

        MIN(H_Points), SUM(H_Points)

FROM House

WHERE          H_Points LIKE '7%';
```

10)

Display The Maximum CGPA Of The Students By Grouping Their House Number

=

```
SELECT   H_No, max(S_CGPA)

FROM     Student

GROUP BY H_No;
```

11)

Display The Greater Average CGPA Than 3.5 Of the Students by Grouping Their House Number

=

```
SELECT   H_No, AVG(S_CGPA)

FROM     Student

GROUP BY H_No;

HAVING        AVG(S_CGPA) > 3.5;
```