**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 Table: House**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

|  |  |  |
| --- | --- | --- |
| 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

--------User Creation--------

CREATE USER Dumbledore IDENTIFIED BY Phoenix DEFAULT TABLESPACE users QUOTA UNLIMITED ON users;

GRANT CREATE SESSION, CREATE TABLE TO Dumbledore;

CONNECT Dumbledore/Phoenix;

--------Table Creation--------

CREATE TABLE House (

H\_Id NUMBER PRIMARY KEY,

H\_Name VARCHAR2(50) NOT NULL,

H\_Points NUMBER

);

INSERT INTO House (H\_Id, H\_Name, H\_Points)

VALUES (11, 'Gryffindor', 892);

INSERT INTO House (H\_Id, H\_Name, H\_Points)

VALUES (22, 'Hufflepuff', 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);

CREATE TABLE Student (

S\_Id NUMBER PRIMARY KEY,

S\_Name VARCHAR2(50),

S\_CGPA NUMBER(3, 2),

S\_BloodStatus VARCHAR2(50),

H\_No NUMBER,

CONSTRAINT cgpa\_check CHECK (S\_CGPA <= 4.00),

CONSTRAINT house\_fk FOREIGN KEY (H\_No) REFERENCES House(H\_Id)

);

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, 33);

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);

--------SQL Queries--------

1. Using ARITHMETIC operator:

Question: Find the average CGPA of all the students with two decimal point.

Answer: SELECT ROUND(AVG(S\_CGPA), 2) AS Average\_CGPA FROM Student;

2. Using CONCATENATION operator:

Question: Retrieve the full name of each student by concatenating their first name and blood status.

Answer: SELECT S\_Name || ' (' || S\_BloodStatus || ')' AS Full\_Name FROM Student;

3. Using COLUMN ALIAS:

Question: Display the blood status of each student with a column alias "Blood\_Status".

Answer: SELECT S\_BloodStatus AS Blood\_Status FROM Student;

4.Using LIKE operator:

Question: Retrieve the names of all students whose names start with "H".

Answer: SELECT S\_Name FROM Student WHERE S\_Name LIKE 'H%';

5.Using IS NULL operator:

Question: Retrieve the students who do not have a blood status specified.

Answer: SELECT \* FROM Student WHERE S\_BloodStatus IS NULL;

6. Using ORDER BY clause:

Question: Retrieve the students' names and CGPAs in ascending order of CGPA.

Answer: SELECT S\_Name, S\_CGPA FROM Student ORDER BY S\_CGPA ASC;

7.Using SUBSTR function:

Question: Retrieve the first three characters of the names of all students.

Answer: SELECT SUBSTR(S\_Name, 1, 3) AS First\_Three\_Characters FROM Student;

8.Using NVL function:

Question: Retrieve the blood status of each student, replacing any NULL values with "Unknown".

Answer: SELECT NVL(S\_BloodStatus, 'Unknown') AS Blood\_Status FROM Student;

9.Using MAX function:

Question: Find the highest CGPA among all the students.

Answer: SELECT MAX(S\_CGPA) AS Highest\_CGPA FROM Student;

10. Using SUM function:

Question: Calculate the total points earned by all the houses.

Answer: SELECT SUM(H\_Points) AS Total\_Points FROM House;

11. Using GROUP BY clause:

Question: Calculate the average CGPA for each house.

Answer: SELECT H\_No, AVG(S\_CGPA) AS Average\_CGPA FROM Student GROUP BY H\_No;

12. Using HAVING clause:

Question: Find the houses with an average CGPA higher than 3.5.

Answer: SELECT H\_No, AVG(S\_CGPA) AS Average\_CGPA FROM Student GROUP BY H\_No HAVING AVG(S\_CGPA) > 3.5;