Data and Applications
Homework 1
Harry Potter

Rohit Gowlapalli
2021101113

Abhinav Reddy Boddu
2021101034

Gnana Prakash Punnavajhala
2021111027

1. **INTRODUCTION:**

The mini world, "**Hogwarts School for Witchcraft and Wizardry**" (short for Hogwarts), is home to both exotic magical items and wizards with unfathomable magical prowess. It comprises of the Hogwarts Castle, the Quidditch playground, the Dark Forest, and many more magical wonders.

2. PURPOSE:

The purpose of the database is to allow the administration to maintain and manage the database and control access to data. It stores details of Teachers, Houses, Students, Game fixtures and other relevant information. In the data requirements section, we show how entities and functions are meaningfully organized to achieve a set of objectives. The database also supports multi-user interface, which enables multiple departments to access and modify data simultaneously.

3. USERS OF THE DATABASE:

1. ***Administration department:***

   These **parametric** end users monitor and formulate the records of students, teachers, and other such users.

2. ***IT department:***

   These **sophisticated** end users implement their own complex requirements to translate the data in the database and maintain the database.

3. ***Accommodation department:***

   These users assign dormitory rooms along with roommates to every student.

4. ***Events department:***

   These users organize and conduct events at Hogwarts.

5. ***House department:***

   These **casual** end users would require continually tracking the progress of their team in the annual Inter-house championship.

6. ***Students:***

   These **stand-alone** users maintain a personal DB using ready-made programs to track their own academic records (reward points, etc.)

4. APPLICATIONS:

The designed database helps the administrators and organizers manage, store, and retrieve data from the details of the students, and establish meaningful connections between varied data.

For instance, the database could enable one to:

- **Get the Quidditch tournament fixtures.**
- **Calculate the number of points a student contributed to a house during the academic year.**
- **Get Total points rewarded to a House till any point of time.**
- **Generate a leader board of students according to their reward points.**

5. DATABASE REQUIREMENTS:

1. ***Entity Types*:**

Entity Type	Attribute	Attribute Type	Sub Attributes	Data Type	Constraints
STUDENT	Name	Composite Key	first_name	VARCHAR	Upto 50 characters
			last_name	VARCHAR	Upto 50 characters
	House points owned	Simple	-	INT	Upto 3 digits

	Spells Learnt	Multi-Valued	-	VARCHAR	Upto 50 characters
	Age	Simple	-	INT	Any 2 digits
	Year of study	Derived attribute. (Derivable from Age)	-	INT	Any 4 digits
	Magical Items owned	Multi-valued	-	VARCHAR	Upto 50 characters
HOUSE:	House Name	Key attribute	-	VARCHAR	Upto 50 characters
	Colour	Simple	-	VARCHAR	Upto 50 characters
	Symbol	Simple	-	VARCHAR	Upto 50 characters
TEACHER	Name	Composite Key	first_name	VARCHAR	Upto 50 characters
			last_name	VARCHAR	Upto 50 characters
	Age	Simple	-	INT	Any 3 digits
SUBJECT	Subject_Code	Key attribute	-	VARCHAR	Upto 50 characters
	Magical_tools_used	Multivalued	-	VARCHAR	Upto 50 characters
QUIDDITCH MATCH	Date	Simple	-	DD/MM/YYYY	10 characters
	Time	Simple	-	HH:MM	5 characters
	Match_number	Key attribute	-	INT	Upto 2 digits
	Winning House	Simple	-	VARCHAR	Upto 50 characters
	Losing House	Simple	-	VARCHAR	Upto 50 characters

2. ***Weak Entities***:
   a. **PET**:
   PET is a **weak entity** having **identifying relationship** with **STUDENT**.
      i. Animal (cat/owl/toad)
         - **Partial Key attributes**

3. ***Relationship Types:***

Relationship Type	Degree	Entities	Cardinality Ratio	Constraints
BELONGS_TO	2	*Student* BELONGS_TO *House*	N:1	STUDENT (1, 1) HOUSE (1, N)
LEADS	2	*Teacher* LEADS *House*	1:1	TEACHER (0, 1) HOUSE (1, 1)
OWNED_BY (Identifying Relationship type)	2	*Pet* OWNED_BY *Student*	1:1	PET (1, 1) STUDENT (0, 1)
TEACHES … TO	**3 (Ternary)**	*Teacher* TEACHES *Subject* TO *Student*	1:1:N	TEACHER (0, 1) : SUBJECT (0, 1) SUBJECT (0, N) : STUDENT (1, M) TEACHER (0, N) : STUDENT (1, M)
PLAYED_BY	2	*Quidditch Match* PLAYED_BY *House*	1:2	QUIDDITCH MATCH (2,2) HOUSE (0,N)

6. FUNCTIONAL REQUIREMENTS:

- The contents of the database are accessible to each end user unless they are obtainable only through those query types that are of type ***[ADMIN]***, in which case, it can be accessed only by the administrators

1. **Modifications:**
   - **a.** ***Insert [ADMIN]*:**
     - i. **insert_student:** inserts a student into the Database
     - ii. **create_match:** creates a new Match in the Quidditch Tournament
   - **b.** ***Update [ADMIN]*:**
     - **i. update_house_points:** updates the total reward points allotted to a House
     - **ii. deduct_student_points:** deducts points of student in case of violation of the code of conduct
   - **c.** ***Delete [ADMIN]:***
     - i. **delete_student:** deletes the record of an outgoing student from the database
2. **Retrievals:**
   - ***a.*** ***Search:***
     - i. **search_student:** Search for student by student_name in student table of the database.
   - b. ***Sorting*:**
     - i. **sort_by_student_name:** sorts the list of students in the database in lexicographical order of their names.

ii. **sort_by_house_points [ADMIN]:** sorts the list of houses in the database in lexicographical order of their total points.

c. ***Get (access)*:**

i. **get_house_points_by_name [ADMIN]:** get the Total reward points of the house corresponding to the House name

ii. **get_quidditch_fixtures:** get the scheduled matches for Quidditch Tournament

iii. **get_student_grades_by_name:** get the grades of a student given their name

d. ***Reports [ADMIN]:***

i. **get_Quidditch_tournament_report:** Display results of all the games held in the Quidditch tournament of that year

ii. **get_event_feedback_report:** Show all the feedback given regarding the event to the event organisers.

7. SUMMARY:

We have given the requirements to implement the database that stores data about the students studying at Hogwarts, Professors teaching at Hogwarts, events like Quidditch, and so on. The database provides functionalities like inserting, updating and deletion of data to those with admin privileges and functionalities like getting student name and quidditch fixtures to everyone. The database also facilitates getting feedback of events and Quidditch matches to administrators. It also enables them to view the list of houses sorted in the order of their points, while a similar sorting functionality which displays the list of students sorted according to their names is provided to everyone.

We have appropriately provided the list of entity types, both strong and weak entity types, their corresponding attributes, of the types, simple, composite, key, partial key, derived and multivalued attributes, and the relationship types of the relations between these entity types, which are of both normal and identifying relationship types, with their appropriate degrees and cardinality ratios and constraints.