Team RBC

REQUIREMENTS DOCUMENT

The goal of this project is to provide a detailed database for easy administration of a MESS at a university. Each student is supposed to eat in a mess for a whole month and only can change a mess at the start of each month(Dependents and visitors of the students will eat in the mess assigned to the students).An account for all the students is generated to pay mess bill as there is no constraint on the number of cancellations and to maintain the messes in a smooth way without incurring a loss by increasing the prices periodically . Details about the mess and the number of cancellations will be updated every month.

When a student enters the university he/she will be randomly assigned a default mess and every month he has an option to change the mess if he/she wants to eat in a mess other than the mess he ate in the previous month.

DATABASE REQUIREMENTS

1)Entity Types:-

a)Student - Details of a student enrolled in a university.
b)Mess - Details about a mess which a student can choose
c)MessMaintenance : Details about maintenance of the mess
d)Employee - Details of people working for different messes
e)Cancellations - Number of cancellations taken by a student in a particular month
f)Accountants - Calculating and maintaining mess bills of various students for all messes.
g)Dependents - Family members of students who can avail certain facilities at the MESS

Entity Type - STUDENT

1)Student Name (VARCHAR(50),NOT NULL) : The name of the student
-Composite attribute containing First name & Last name
2)Gender(CHAR(),NULL): The gender of the student
-Simple Attribute {M,F}
3)StudentId(INT(10),NOT NULL) : ID of the student
-Key Attribute
4)Batch and Programme(VARCHAR(20),NOT NULL) : Joining batch and programme opted
-Composite Attribute
-Each sub part has sub classes
-Batch has 2k19,2k18,2k17,2k16,2k15,2k14 subclasses(assuming year 2k19 as present year and will be updated every year)
-Programme has CSE,ECE,CLD,CSD,ECD subclasses
5)Email Id(VARCHAR(50),NULL) : Email Id of a student
-Simple Attribute
6)Phone Number(VARCHAR(50),NULL) :Phone Number of a student
-Multi Valued Attribute :- One or more phone numbers for communication
7)MessId(INT(5),NOT NULL):The id of mess in which student registered.
-Foreign key

Entity Type - MESS

1)MessName(VARCHAR(50),NOT NULL):Name of the mess

-Simple attribute

2)MessId(INT,NOT NULL): The ID of the mess

-key attribute

3)MessTimings:(VARCHAR(500),NOT NULL) The timings of mess

-Composite Attribute { breakfast & lunch & dinner(open & close)}

4)MessMenu(VARCHAR(5000,NOT NULL): The menu of the given mess every week

-Nested Composite Attribute{Every day { breakfast & lunch & dinner(menu)} }

Entity Type - MESSMAINTAINCE (WEAK ENTITY)

1)MessId(INT,NOT NULL):Id of the mess for which we store details.

-Foreign Attribute

2)NoOfRegistrations(INT,NOT NULL): No of students registered for monthly in that mess

-Derived Attribute from the count of rows in the student of the MessId

3)MessExpenditurePerDay(INT,NOT NULL): Total amount spent per day on easy

-Simple Attribute

4)MessCostPerDay(INT,NOT NULL): The amount to be paid per day

-Simple Attribute

Entity Type - EMPLOYEE

1)EmployeeName(VARCHAR(50),NOT NULL) : The name of the employee

-Composite attribute containing First name & Last name

2)GenderEmployee(CHAR,NULL) : The Gender of the employee

-Simple Attribute

3)EmployeeId(INT,NOT NULL): Unique Id of the employee
-Key Attribute
4)MessId(INT,NOT NULL) : Present id of the mess where the employee works
-Foreign Key
5)Work(VARCHAR(20),NOT NULL) : Kind of job he does in the assigned mess
-Contains Subclasses - Chef, Server, Cleaner, Manager
-Disjoint set of classes
6)Salary(INT,NOT NULL): Salary of the employee
-Simple Attribute

Entity Type - CANCELLATIONS (WEAK ENTITY)

1)StudentId(INT,NOT NULL) : Id of the student whose leaves are stored in the particular row
-Foreign Key
2)LeavesTakenThisMonth(INT,NOT NULL) : Leaves Taken in this month
-Simple Attribute

Entity Type - ACCOUNTANTS

1)AccountantId(INT,NOT NULL) : Id of the accountant
-Key Attribute
2)StudentId(INT,NOT NULL) : The id of the student whose accounts has to checked
- Foreign Attribute(can be multi-valued)

Entity Type - DEPENDENT (WEAK ENTITY)

1)StudentId(INT ,NOT NULL) : Id of the student related to the
-Foreign Key
2)DependentName(VARCHAR(50),NOT NULL) : Name of the dependent

-Simple Attribute

3)DependentGender (CHAR , NULL): Gender of the dependent

-Simple Attribute{M,F}

4)Relationship(VARCHAR(50),NULL) : How the dependent is related to the Student

-Simple Attribute

5)NumberOfDaysEaten(INT , NOT NULL) : How many days the dependent has eaten in the mess for bill calculation

-Simple Attribute

2)RELATIONSHIP TYPES :

1) Eats_in : Student Entity type is in one to many relationship with Mess Entity type .
Where each student can eat in only one mess,but a mess can serve for multiple students.

Can be read as Student X **eats in** Mess Y.

2)**Cancelled** : Cancellation entity type has 1 to 1 relationship with Student entity type as no of cancellations for each student is unique .

Can be read as Student X has cancelled for Y no of days.

3)**IsRelatedTo** : Student entity type has many to one relationship with Dependent entity type as each student can have many dependents,but a dependent is related to only one student.

Can be read as Student X is related to Dependent Y.

4)**WorksFor**: Employee entity type has one to many relationship with Mess entity type as the number of employees working in a particular mess can be greater than 1,but an employee works in a particular mess .

Can be read as Employee X works in Mess Y.

5)**Maintenance** : MessId in the mess maintenance entity type has one to one relationship with MessId in the mess entity type.

Can be read as Mess X maintenance details are there in MessMaintenance Y.

6)**BillCalculation** : Accountant entity type has one to one relationship with student entity type and student entity type has one to many relationship with mess maintenance entity type ,Accountant entity type has many to one relationship with dependent entity type ,Accountant entity type has one to one relationship with Cancellations entity type .

Can be read as Accountant X calculates bill of the student Y by using no of cancellations in a month from cancellations entity type and Accountant X also gets MessCostPerDay using mess id of a student , Accountant X also considers no of days dependent eat in that month to calculate monthly bill of the student.

FUNCTIONAL REQUIREMENTS

1)LOADING OR INSERTION OF DATA

i)When a new student enters the university his data would be inserted in the student entity type.
ii)When a new dependent enters a university to meet the student his/her data will be added to the Dependent's table.
iii)A new mess can be added to the list of messes if required.

2)DELETION OF DATA

i)After a student graduates his data from the database will be deleted
ii)If an employee resigns ,his information can be removed

3)MODIFICATION AND UPDATION OF DATA

i)After every month the number of leaves attribute in cancellations database will be set to zero

ii)After every month the number of days eaten by the dependent will be set to zero

iii)If a student cancels food for a day the count increases in leaves taken And hence the bill generated will be dynamically changed similarly for Dependent number of days eaten.

iv)Mess Menu can be updated monthly if required .

v)List of all possible batches will be updated every year.

vi)Mess where an employee works can be updated every month.

4)REPORT GENERATION FROM DATA

i)Total profit or loss can be calculated for each mess as well as the university based on total payment received from students and total salaries given to employees with expenditure per month.

ii)Total bill of the student in the year can be calculated from his /her monthly bills.

iii)No of students registered in a mess can be calculated from mess id's of all students. This can add a new attribute to mess entity type.

iv)Average number of cancellations of students can be calculated.