

Project Step 2 Draft

Josiah Herrera

Khrystian Clark

A) Fixes from Step 1

- Ethan Feedback:
 - **Feedback:** “The datagram gives a basic overview, but besides the names of the entities and the relationships, none of the attributes were filled in.” – **Action:** we added attributes in
 - **Feedback:** “I think LearningResource should become Learning_resource to fit better” – **Action:** we fixed this
- Cheyenne Feedback:
 - **Feedback:** The overview “notes that tracking of tuition and learning resources will be available, but not explained how and in what relationships.” **Action:** We will not act on this in the overview since it’s the purpose of the outline and ERD to specify the how, which we believe are clear.
 - **Feedback:** “For this table, specifically the Days_available, maybe creating a schedule table that holds the types of schedules available to the teachers. For example, sched_1 - M,W,F or sched_2 T,R. With sched_id being the PK and having additional attributes of Monday, Tuesday, Wednesday, etc. being bool. Then utilizing sched_id as an FK in the employee table.” **Action:** We will not act on this. The business model of the daycare has changed. The daycare is now Monday-Friday and all employees are available on those days. Thus, we are removing this attribute.
 - **Feedback:** “I would also suggest utilizing the money or decimal data type for any monetary attributes that you have” – **Action:** We have updated this data type to be decimal(6,2)
 - **Feedback:** “There are 1:M relationships, but one is not formulated correctly, being the relationship between Classes and Children” – **Action:** We have updated this relationship to where the child links to a class only
 - **Feedback:** “Partially, there is some consistency, but what is described in the overview is not fully represented in the relationships on the ERD.” – **Action:** we are reviewing, but this feedback is not specific enough to action on.
- Benjamin Feedback:
 - **Feedback:** “it's not clear how the website will keep track of tuition and payment status, or learning resource inventory.” **Action:** – we added some additional information in the overview about the types of webpages we will have for the end user to keep track of this information.
 - **Feedback:** “It would be useful to know how many applications the daycare processes. More details about the learning resources would also be helpful” – **Action:** We have added more details about the number of applications and the types of learning resources available.

- **Feedback:** “None of the attributes in the Classes table is related to the age of the children” – **Action:** added a field for this
- **Feedback:** “there does not seem to be any constraint on the Classes related to a maximum of two classes for each age group” – **Action:** The use case has been updated to not have this constraint as real estate for additional class rooms is readily available.
- **Feedback:** “For the Employees table, it's not clear what the credential table will do” – **Action:** We are taking this reference to credential out. It was a former idea that we abandoned
- **Feedback:** “For the LearningResource table, it's not clear what the purpose of the table is. It is clear that classes can have learning resources and that a child can use one at a time, but it's unclear what the purpose of these relationships are.” – **Action:** we added a sentence in the outline of the Learning_Resources describing its purpose
- **Feedback:** “The primary key in each table is identified as unique, however, it's not necessary to do this since primary keys are unique by default” – **Action:** we have taken out this unnecessary specification.
- **Feedback:** “In the Classes table, cost_per_child should likely have a constraint given that tuition is between \$800-1200 as described in the overview” – **Action:** We have added this constraint. Note that it makes more sense to add it to the data entry point in the web app as opposed to the database.
- **Feedback:** “In the Employees table, Days_available might be better implemented as a many to many relationship with an intersection table between Employees and a category table that lists the days of the week” – **Action:** We will not act on this. The business model of the daycare has changed. The daycare is now Monday-Friday and all employees are available on those days. Thus, we are removing this attribute.
- **Feedback:** “Under the Children table, Payment_status might be better implemented as a boolean (tinyint) attribute, such as "tuition_is_paid". **Action:** We have updated to tinyint
- **Feedback:** “It would be helpful to list each relationship as a separate bullet point.” – **Action:** - we adjusted this to be separate bullets
- **Feedback:** “The relationship between Classes and Children is identified as being implemented with a foreign key inside of the Employees table, but if Children are assigned to a teacher and not a class, then this is a relationship between Children and Employees, not Children and Classes.” – **Action:** We are not acting on this. In our use case, children and teachers are assigned to classes. There is no direct association between children and teachers, only indirect through classes being a bridge.
- **Feedback:** “It's not clear why the relationship between LearningResources and Classes is implemented with a foreign key in Children. If Classes can have many LearningResources and LearningResources can be used in many classes, then that relationship should be implemented with an intersection table between LearningResources and Classes.” – **Action:** This is a valid comment, but we chose to not model an intersection table in our ERD since it's not strictly necessary in an ERD. We will have an intersection table between classes and learning resources and will add this to the ERD to reduce confusion.

- **Feedback:** “In Employees, only one relationship is defined, which is between Classes and Employees, but the description below notes that a teacher can have many children and many learning resources, so those relationships should be defined with an intersection table between Employees and Children and Employees and Learning Resources” – **Action:** We have made it so that there is no direct relationship between children and teachers. Instead, it goes through class, which was our original intent.
- **Feedback:** “As mentioned in the previous question, the 1:M are not formulated correctly. In the Classes table, a 1:M relationship between Child and Classes is implemented with a foreign key in Employee. However, if a class can have many children, but a child can only have one class, this should be implemented with a foreign key in Children, which identifies which class that child is assigned to” – **Action:** we fixed this by adding a class_id in children
- **Feedback:** “in the Classes table, the M:M relationship between LearningResources and Classes is implemented with a foreign key in Child. However, many to many relationships are implemented with an intersection table between the two tables/entities that have a many to many relationship.” – **Action:** we added intersection table to reduce confusion
- **Feedback:** “The tables that will represent each entity are not referred to in the overview” – **Action:** No action – this is not required in the overview.
- **Feedback:** “Other than Classes, the names of the tables are not pluralized, but they are capitalized” – **Action:** we have fixed this
- **Feedback:** “Relationships between Children and Employees and Children and Classes is described in the entities table section. In the ERD no relationship between Children and Employees is noted. So consider deciding how best to implement the relationships between those three entities.” – **Action:** No action, we will connect employees with children through the class object.
- **Feedback:** “Provide a more detailed description of what a LearningResources is and what the purpose of tracking them is. Is there a row for each individual (for example) pencil and toy? If not, then there should be an attribute in that table such as "total_inventory" or "total_on_hand" that tracks how many the school has since the description says this entity will help with tracking inventory and keeping track of purchasing status.” – **Action:** We added a field “On_hand_qty” to indicate that there is one row per item type, not one row per item instance
- Donald Feedback:
 - **Feedback:** “Consider including the domains of each data type (e.g., varchar (45)) for your Final Draft” – **Action:** we have updated our varchars to be varchar(50)
 - **Feedback:** “the Cost_per_child attribute should not be a float. Floats shouldn't be used for monetary values as they lack decimal precision. Instead, use decimal/numeric for money.” – **Action:** we have updated to decimal(6,2)
 - **Feedback:** “consider changing the data type of Days_available to something other than varchar. Using a varchar here introduces the possibility for human error into your data. Since there are only two options ("Mon-Thurs" or "Tue-Fri"), you could use a tinyint, for instance, that acts as an enumeration.” – **Action:** We will not act on this. The business

model of the daycare has changed. The daycare is now Monday-Friday and all employees are available on those days. Thus, we are removing this attribute.

- **Feedback:** “consider changing the Payment_status to something other than varchar (like BIT, which is good for represent boolean data)” – **Action:** We have changed it to tinyint
- **Feedback:** “Also, how do you plan to match children to teachers by tracking their ages? It isn't clear to me based on your overview how the Age attribute in Child will be used to sort children into Classes” – **Action** we have added more information to the overview as to how the website will be used
- **Feedback:** “the item_price attribute should not be a float, since it is a monetary value.” – **Action:** we have updated the data type to be decimal
- **Feedback:** “All relationships are clearly formulated, except for maybe one. The description of each relationship is thorough, and includes details like which entity in a relationship holds the relevant foreign key. There is a M:M relationship described between LearningResources and Classes, but there is no foreign key described in either entity that shows this relationship. Based on the description, it seems this relationship is derived from the 1:M relationship LearningResource has with Child and the M:1 relationship Child has with Classes (via the Teacher_id FK in Child, connecting many children to a single teacher, who is then connected to Classes in a 1:1 relationship), but I'm not sure that this actually translates to a M:M relationship between LearningResource and Classes... if I'm wrong and this does indeed make a valid M:M relationship, I think it could at least be made more simple with a foreign key connecting LearningResource with Classes directly” – **Action:** We have added an intersection table to make this more explicit
- **Feedback:** “there is an inconsistency between the relationship described in Child and how that relationship is represented in the diagram: a M:1 relationship is described to exist between Classes and Child via the teacher_id FK in Child connecting each Child with an Employee, but this relationship is not shown in the ERD. Your diagram doesn't show a relationship between Child and Employee, despite the FK in Child establishing a relationship between those entities.” – **Action:** We have removed the relationship between child and teacher. Instead, this relationship will be bridged through class.

Additional Updates

- In addition to the actions listed above, we have made the following edits to the design:
 - We have changed the “Employees” entity to be “Teachers” and we have updated all references since we are really only concerned with teachers and not other ancillary employees
 - We have changed the naming of fields to be snake case with the first letter capitalized to be consistent
 - We added a primary key to the intersection table that is not a composite key in order to follow the best practice outlined in the class

- We added a Class_Costs table since the cost of the class is solely determined by the age range. Thus in order to be 3nf, we broke this out.

B) Project Outline and Database Outline – Updated Version

Project title: “Dads with Babies”

a) Overview (one paragraph)

- a. Daddy’s Daycare offers childcare for ages 0-5, for a monthly payment that ranges between \$800 to \$1200 (depending on the age of the child). Classes are split by age group (0-1, 1-2, 2-3, 3-4, & 4-5) with one teacher per class. Each class has a max capacity of 4 children and each teacher takes on only one class. A database driven website will help match each of the customer’s children with one of the available classes by efficiently tracking their ages and capacity of the classes. Tracking includes information about the child, and the teachers. This database will enable us to strategize where to place new children (roughly 4 new applicants a month), how close we are to capacity, and revenue/tuition payment statuses. It will also help us track inventory on learning resources (books, toys, etc.) for the children, purchasing status of the items, and inform future purchases based on what items are the children’s favorites. We will accomplish this via webpages that both allow the daycare employee to add and update information as well as see the status and relationships between existing information, such as how many students a class has, how many more slots available we have for a particular age group, how much inventory of a particular item we have, which students have not paid their tuition etc.

b) DB outline (bullet format, 4 entity tables, 4 relationships)

a. **Classes:** This is any of the classes, as separated by age in years.

- Class_ID: int, not NULL, PK, auto_incrementing
- Class_Name: varchar (50), not NULL
- Class_Cost_ID int, not NULL, FK
- Class_Size: int, NULL if there aren't any children enrolled, max 4
- *Relationship(s):*
 - 1:1 relationship between Teachers and Classes with Classes (Class_ID) as FK inside of Teachers.
 - 1:M relationship between Classes and Children with Classes (Class_ID) as FK in Children.
 - M:M relationship between LearningResource and Classes – implemented as an intersection table (Classes_Learning_Resources_Intersection) where there is a 1:M relationship between classes and the intersection table
 - A M:1 relationship between Classes and Class_Costs
- *A class has a single age range, one teacher, up to 4 children, one cost ID, and 0 or many learning resources*

b. **Teachers:** This contains the full list of teachers available. Teachers must have a class field for activation.

- Teacher_ID: int, not NULL, PK, auto_incrementing
- First_Name: varchar (50), not NULL
- Last_Name: varchar (50), not NULL
- Class_ID: int, FK, not NULL
- *Relationship(s):*
 - 1:1 relationship between Classes and Teachers with Classes (Class_ID) as FK in Teachers.
- *A teacher is only ever associated with one class*

c. **Children:** This contains information on the children currently enrolled, with reference to class, age, favorite Learning Resource and tuition paid status

- Child_ID: int, not NULL, PK, auto_incrementing
- First_Name: varchar (50), not NULL
- Last_Name: varchar (50), not NULL

- Age: int, not NULL
- Class_ID: int, FK, not NULL
- Tuition_Is_Paid: tinyint, (true or false)
- Favorite_Learning_Resource_ID: int, FK, not NULL
- Relationship(s):
 - M:1 relationship between Child and their favorite Learning Resource with Favorite_Learning_Resource_ID (Learning_Resource_ID) as FK in Child.
 - M:1 relationship between Classes and Child with Teachers (Class_ID) as FK in Child
- *One child can have one class and one favorite learning resource.*

d. **Learning_Resources:** List of learning resources and educational toys available at the daycare. A child's favorite learning resource is tracked via child interaction observations made by the teachers and classes can have the same learning resource as other classes and multiple learning resources at once. The purpose of these relationships will be so that the teachers and daycare admins can track which resources are favorites of students and make sure they are available in the appropriate classes.

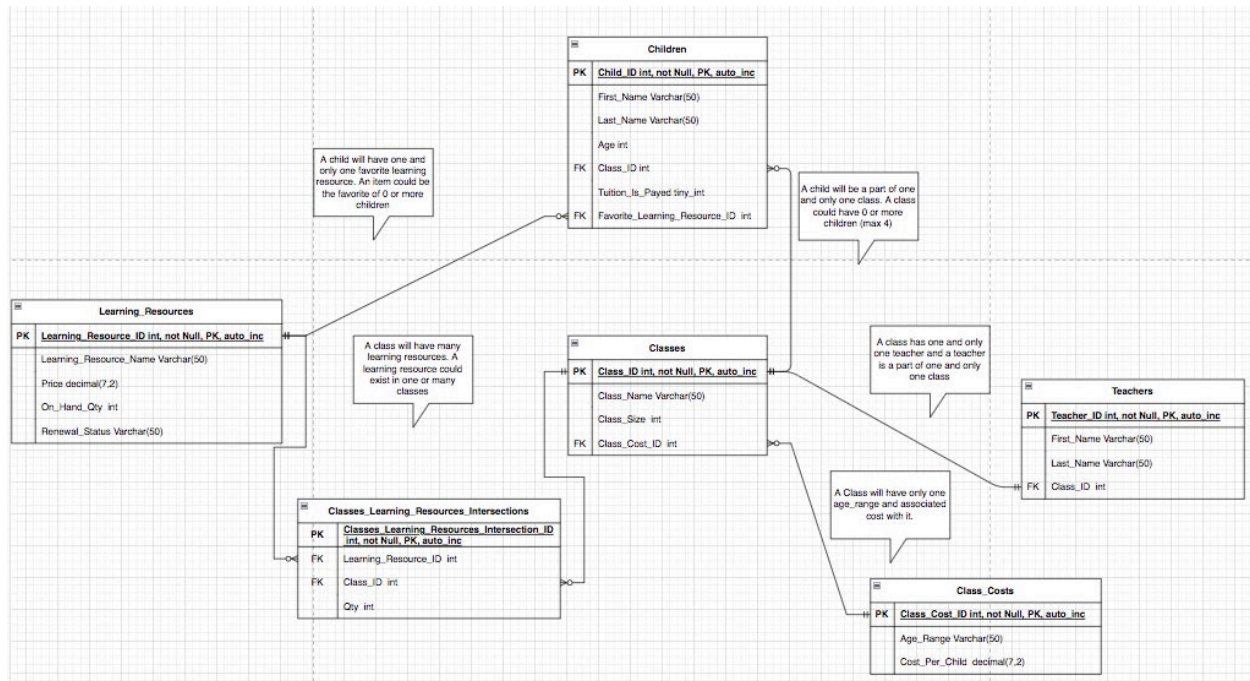
- Learning_Resource_ID: int, not NULL, PK, auto_incrementing
- Learning_Resource_Name: varchar (50), not NULL
- Price: decimal (7,2), not NULL
- On_Hand_Qty: int, not NULL
- Renewal_Status: varchar (50), not NULL (for subscription-based items)
- Relationship(s):
 - M:M relationship between Classes and LearningResources implemented as an intersection table (Classes_Learning_Resources_Intersection) where there is a 1:M relationship between Learning_Resources and the intersection
 - M:1 relationship between Child and LearningResource with Favorite_Learning_Resource_ID (Favorite_Learning_Resource_ID) as FK in Child
- *One Learning resource can be associated with many classes and many children as their favorite learning resource*

e. **Classes_Learning_Resources_Intersections:** This is an intersection table that lists out the intersections in the M:M relationship between Learning_Resources and Classes. As such, its primary key is a composite primary key (the combination between the Learning_Resource_ID and the Class ID). This table will allow us to see the amount of learning resources in each class as well as what they are, which will help teachers and administrators make inventory decisions

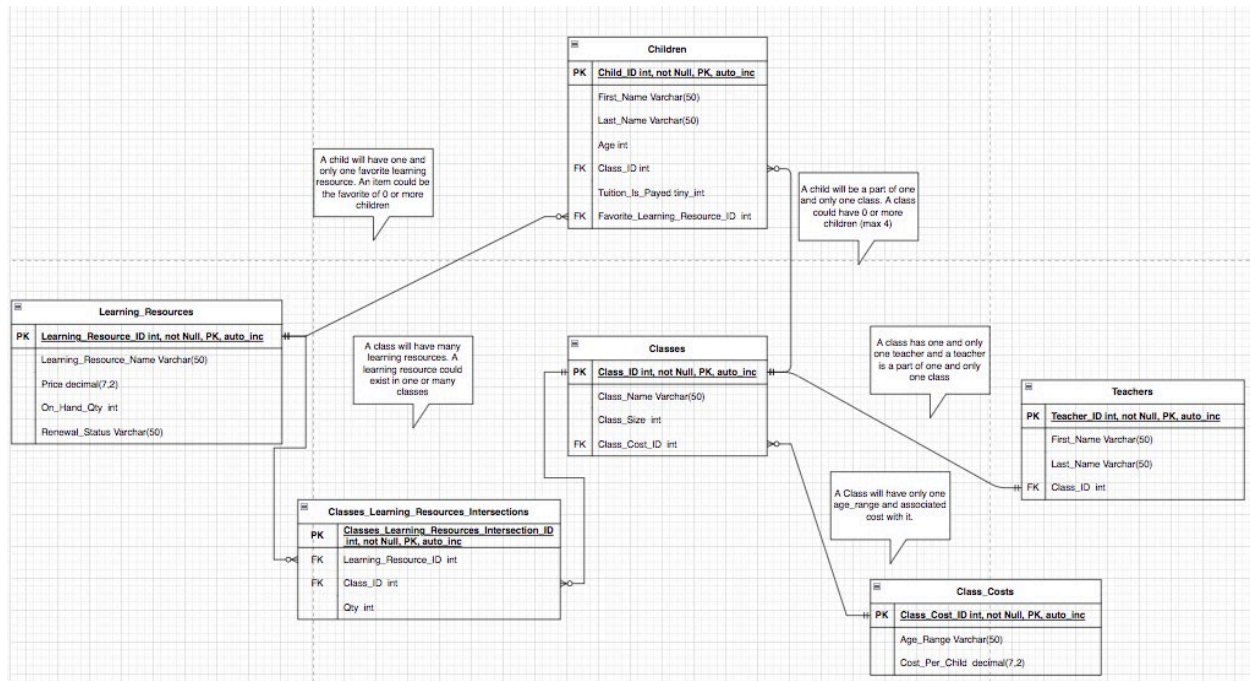
- Class_Learning_Resource_Intersection_ID: int, not NULL, PK, auto_inc
- Learning_Resource_ID: int, not NULL, FK
- Class_ID: int, not NULL, FK
- Qty: int, not NULL

- *Relationship(s):*
 - 1:M relationship between Classes and Classes_Learning_Resources_Intersections
 - 1:M relationship between Learning_Resources and Classes_Learning_Resources_Intersections
 - *One Learning Resource can be associated with many classes (since we can have more than one of the same resource) and many classes can utilize the same Learning Resource*
- f. **Class_Costs:** This is an attribute table that keeps track of the costs associated with an age range since an age range will always have the same cost.
- Class_Cost_ID int, not NULL, PK, auto_incrementing
 - Age_Range: varchar (50), not NULL, constraint on values: '0-1',..., '4-5'
 - Cost_Per_Child: decimal (6,2), not NULL (\$), constraint (value 800-1200)
 - *Relationship(s)*
 1. A M:1 relationship between Classes and Class_Costs
 - *A class cost is associated with the class*

C) Entity-relationship diagram



D) Schema



E) Example Data

Children

Child_ID	First_Name	Last_Name	Age	Class_ID	Tuition_Is_Paid	Favorite_Learning_Resource_ID
1	Devin	Peterson	2	1	1	3
2	Anaka	Herrera	2	1	1	2
3	Gang	Jin	1	2	1	1

Learning_Resources

Learning_Resource_ID	Learning_Resource_Name	Price	On_Hand_Qty	Renewal_Status
1	Where is Waldo	14.99	3	1
2	Rubik's Cube	10.99	2	0
3	Paint Book	29.99	5	1

Classes

Class_ID	Class_Name	Class_Size	Class_Cost_ID
1	Favia's Painters	2	2
2	Joe's Runners	1	1

3	Maria's Readers	0	3
---	-----------------	---	---

Class_Costs

Class_Cost_ID	Age_Range	Cost_Per_Child
1	0-1	800
2	1-2	900
3	2-3	1000
4	3-4	1100
5	4-5	1200

Teachers

Teacher_ID	First_Name	Last_Name	Class_ID
1	Favia	Hegiste	1
2	Joe	Nixon	2
3	Maria	Araujo	3

Classes_Learning_Resources_Intersections

Classes_Learning_Resources_Intersections_ID	Learning_Resource_ID	Class_ID	Qty
1	1	1	2
2	1	2	1
3	3	1	3