1. Scaler will have multiple **batches**.
2. For each batch, we need to store the name, start month and current instructor.
3. Each **batch** of Scaler will have multiple **students**.
4. Each batch has multiple **classes**.
5. For each class, store the name, date and time, instructor of the class.
6. For every student, we store their name, graduation year, University name, email, phone number.
7. Every student has a **buddy**, who is also a student.
8. A student may move from one batch to another.
9. For each batch a student moves to, the date of starting is stored.
10. Every student has a **mentor**.
11. For every mentor, we store their name and current company name.
12. Store information about all **mentor sessions** (time, duration, student, mentor, student rating, mentor rating).

13. For every batch, store if it is an Academy-batch or a

Batch Table

		Instructor Id	Batch Type Id
		1	1
		2	2

.

Batch Type (acts a look up table)

1	ScalerAcademy
2	Data Science

- Saves space
- No need to update every string inside batch table in case batch type changes

Instructor (acts a look up table)

1	Prateek
2	Deepak

1. Create Tables

Batches
Students
Classes
Mentors
MentorSessions

2. Attributes of the tables / Primary Keys

Batches (**batch_id,** name, start_month,instructor_id, batch_type)

Students (**student_id**, name, graduation_year, university_name, email, phone_number,)

Classes (**class_id**, name, date, time, instructor_id)

Mentors (**mentor_id**,name,company_name)

MentorSessions (**session_id**, time, duration, student_id, mentor_id, student_rating, mentor _rating)

Instructors (instructor_id, name)

BatchTypes (batch_type_id, batch_type_value)

Primary Key Expectations:

1) It should be unique & rarely change.
2) It should be fast to sort upon - ints have small size.
3) Naming Convention: batch_id, student_id, mentor_session_id etc.

3. Relationships between the tables

CARDINALITY CONCEPT

Batches (**batch_id,** name, start_month,instructor_id, batch_type)
PK: batch_id
FK: instructor_id
FK: batch_type

Students (**student_id**, name, graduation_year, university_name, email, phone_number, **batch_id, buddy_id,mentor_id**)
PK: student_id
FK: batch_id
FK: buddy_id but in the same table
FK: mentor_id

Classes (**class_id**, name, date, time, instructor_id)

Mentors (**mentor_id**,name,company_name)

MentorSessions (**session_id**, time, duration, **student_id, mentor_id**, student_rating, mentor _rating)
PK session_id
FK student_id
FK mentor_id

Instructors (**instructor_id**, name)
PK: instructor_id

BatchTypes (batch_type_id, batch_type_value)
PK: batch_type_id

StudentBatch
(**student_id, batch_id**, date_of_shift)
PK: (student_id, batch_id) under the assumption that the pair is always unique
FK: student_id
FK: batch_id

Student

s_id	name	email	phone	current_batch
1	prateek	xyz	abc	7
2	deepak	-	-	5

Student Batch -> Sparse Table (very less no of rows compared to student table)

1	5	10 april
1	3	11 june
1	7	18 june

Primary Key Expectations:

4) It should be unique & rarely change.
5) It should be fast to sort upon - ints have small size.
6) Naming Convention: batch_id, student_id, mentor_session_id etc.