VCE APPLIED COMPUTING · UNIT 1 OUTCOME 1 · RELATIONAL DATABASES · START OF LESSON 3

ERD Clinic: four designs that nearly work

Before we build the database, check that the design is right. A mistake in the ERD becomes a mistake in every table, query and test after it.

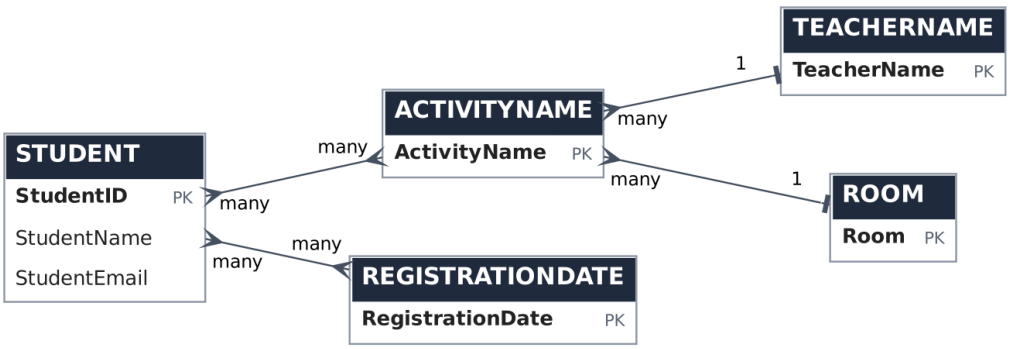
Read this first

Your teacher wrote these four designs to show mistakes people often make with Ridgeview. They are **constructed examples, not real student work**.
Each one would *run*. Your job is to explain what would go wrong when the office starts using it.

Hinge questions (mini-whiteboards or ABCD cards)

1. The office wants to see who runs each activity. Where should **TeacherID** go? **A** in Student · **B** in Activity · **C** in Teacher, stored as ActivityID · **D** in both Teacher and Activity
2. Which is the best primary key for Student? **A** StudentName · **B** StudentEmail · **C** StudentID · **D** StudentName + YearLevel
3. Which SQLite data type should store a phone number such as 0395550142? **A** INTEGER · **B** REAL · **C** TEXT · **D** BLOB

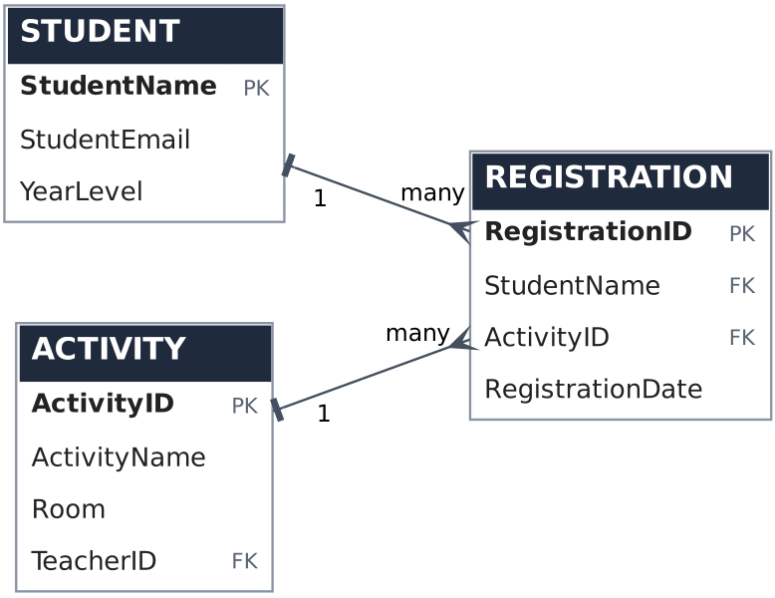
Sample A



This design gives every column its own box. Which boxes are really *attributes*? What has happened to the link between a teacher and the activities they run?

What is wrong, and what would happen in the real database?	Fix it: redraw or describe the change

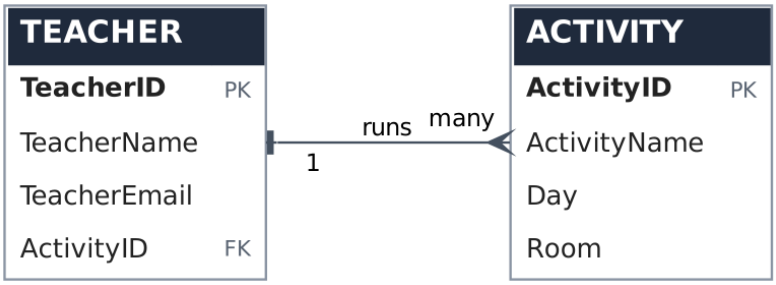
Sample B



Look closely at the keys. What happens when two students called Mei Lin join the same club? What happens when a student changes their surname?

What is wrong, and what would happen in the real database?	Fix it: redraw or describe the change

Sample C



The relationship line says one teacher runs many activities. Read the keys: how many ActivityID values can one Teacher row hold? What happens when L. Okafor takes on a third club?

What is wrong, and what would happen in the real database?	Fix it: redraw or describe the change

Sample D

This Teacher table has the right entity and keys, but look at the data types and the two rows the office entered.

Attribute	Data type chosen	Stored value
TeacherID (PK)	TEXT	T01
TeacherName	TEXT	L. Okafor
Phone	INTEGER	395550142
StartYear	TEXT	2019, then 'twenty twenty-one' for T. Nguyen

The office typed 0395550142 for L. Okafor's phone. Why was it stored as 395550142? What can the database no longer check about StartYear? Choose a better type for each attribute, using only NULL, INTEGER, REAL, TEXT or BLOB.

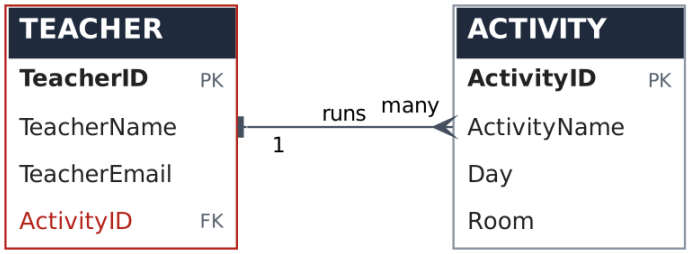
What is wrong, and what would happen in the real database?	Fix it: the type you would choose, and why

Exit check

Finish this sentence about one sample: *This design would cause a problem when ___ because ___. I would fix it by ___.*

Teacher notes

Sample	Misconception	Expected diagnosis and fix
A	Treating every spreadsheet column as an entity (anticipated in the unit plan).	ActivityName, TeacherName, Room and RegistrationDate are attributes, not entities. The design loses the facts that link them (who runs which activity, when a student signed up). Fix: Teacher, Activity, Student and Registration entities; Room inside Activity; RegistrationDate inside Registration.
B	Using a name or email as a primary key (anticipated in the unit plan).	StudentName is not unique and can change; it is also copied into Registration as a foreign key, so a name change breaks every sign-up. Fix: StudentID as PK and FK.
C	Putting the foreign key on the wrong side of a one-to-many relationship (anticipated in the unit plan).	One Teacher row can hold one ActivityID, so the diagram's '1 to many' cannot be stored. Fix: TeacherID as an FK in Activity, the 'many' side.
D	Choosing a type by how a value looks, not what it is used for.	A phone number is an identifier, not a quantity: INTEGER drops the leading zero and would allow arithmetic on it. Fix: TEXT. StartYear is a quantity used for comparisons: TEXT accepts 'twenty twenty-one', so INTEGER (plus a range check) is better. A real address database that stores phone numbers as INTEGER shows exactly this loss: 0299305200 becomes 299305200.



Key version with errors in red (Sample C shown; red versions of A and B are in the image folder).

Hinge answers. Q1 **B**. A links the wrong pair of entities; C is the Sample C error; D suggests 'more links is safer' and leads to contradictions. Q2 **C**. A and B are the Sample B error; D knows keys must be unique but forgets that names change. Q3 **C**. A is the Sample D error; B and D show a student guessing from the list.

How to use. Run the hinge questions first, with every student answering at once. Group by response: students who chose correctly work through all four samples in pairs; others work through the sample that matches their error with you. This follows the unit's decision point: exit-ticket evidence decides who needs re-teaching before implementation.

Resource by Arvin Hay, part of the Ridgeview database unit. Diagrams are drawn from the unit's Ridgeview design and use crow's foot notation and the five SQLite data types. All samples are constructed by the teacher and labelled as such.