Introduction to Management Information Systems

Database Design - 1

Data Resource Management

Database Design

- form groups of about 4 students
- you will need a computer (device)
  - you can use AI to create test data
- in future lessons each group will need to
  - download a database engine called SQLite
  - can use a Mac but has more issues
  - copy files from online
  - copy & develop SQL code into the SQLite

you will make mistakes but this is part of the learning process

Create some test data

for example

prompt to create students

create 10 students with the following data in a table of rows for each student

- Student ID
- Student Name
- Course Enrolled (course code)
- Course Enrolled (course name)
- Registration Date (year, semester, e.g. 2024-2)

Here is some example data for the format (mainly Thai & Chinese students)

652415501 Zhang Wei 888342 Management Information systems 2024-2

follow-up prompt

this is ok for data but there are no courses with more than one student, can you provide a course in both semesters for example, and give details of students that are registered on more than one course, and courses that have more than one student - provide as many rows as needed

Student ID	Student Name	Course Code	Course Name	Registration Date
652415501	Zhang Wei	888342	Management Information Systems	2024-2
652415502	Chen Mei	887231	Business Analytics	2024-2
652415502	Chen Mei	888342	Management Information Systems	2024-1
652415503	Li Hua	881234	Marketing Principles	2024-2
652415503	Li Hua	888342	Management Information Systems	2024-1
652415504	Somchai Prasert	888341	Data Science Essentials	2024-2
652415504	Somchai Prasert	888342	Management Information Systems	2024-2
652415505	Wanchai Suthep	883123	Financial Accounting	2024-1
652415505	Wanchai Suthep	881234	Marketing Principles	2024-2
652415506	Liu Xiaojun	884567	Operations Management	2024-1
652415506	Liu Xiaojun	888341	Data Science Essentials	2024-2
652415507	Kritsada Anan	885678	Human Resource Management	2024-2
652415508	Wang Fang	886345	Supply Chain Management	2024-1
652415508	Wang Fang	888342	Management Information Systems	2024-1
652415509	Napat Siriwat	887654	International Business	2024-2
652415509	Napat Siriwat	888341	Data Science Essentials	2024-2
652415510	Zhang Min	888120	Corporate Strategy	2024-1
652415510	Zhang Min	884567	Operations Management	2024-1
652415511	Li Xiaomei	888342	Management Information Systems	2024-2
652415511	Li Xiaomei	888341	Data Science Essentials	2024-1

each group

- download the test data
- add some more rows
- maybe add group members to courses

think about these questions

- what would happen if you have to register thousands of students on courses at one time?

- would there be any errors?

- how would you find any errors?

- Would this be easy?

- Could you guarantee that you could find 100% of the errors?

think about these questions

- can you add students with an wrong student number?

- can you add courses with an wrong course code?

- can you create duplicate records incorrectly?

- can you register students with an invalid student number, or a wrong course code?

- can you re-register students by mistake?

- we solve these problems by using a database

in relational databases

- we solve these error issues by using keys

- in a table

  - we use a unique key

  - such as a student number

  - called the primary key

Create simple tables

student table

course table

database concepts

- each row in a table, is an

  entity instance

- each column is an

  attribute

- one value for each attribute of each entity instance
  - one value is in one row and one column

a table

based on one object / concept

called an entity

e.g. student

title

student
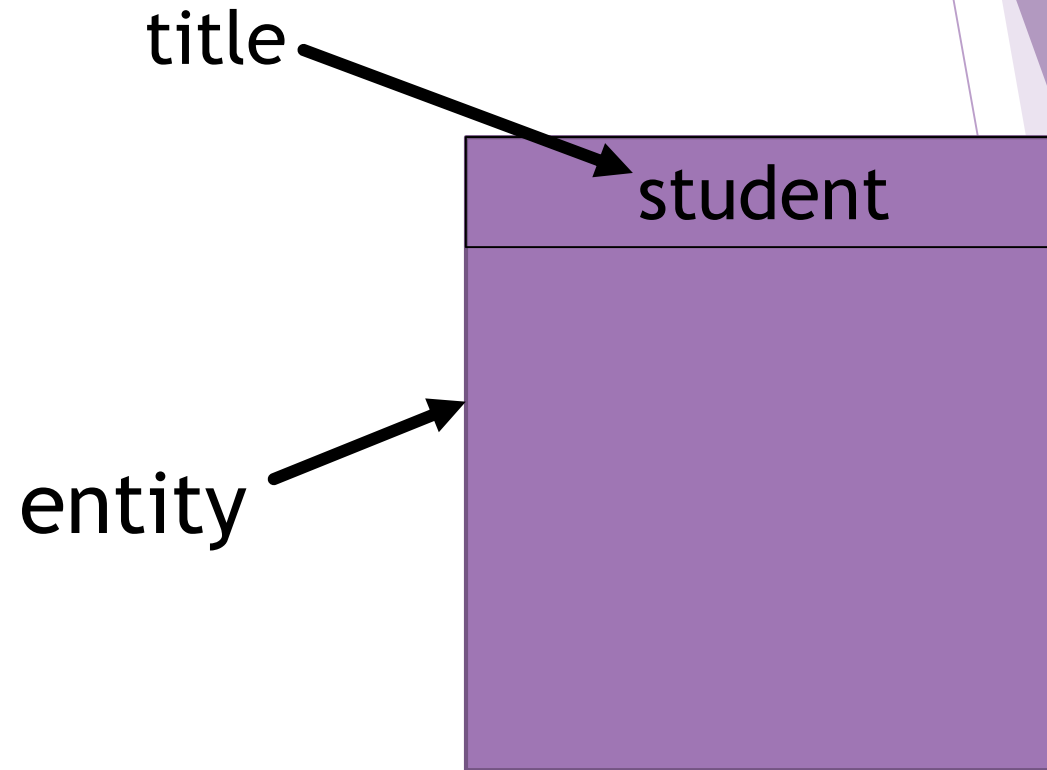
entity

an entity is singular

data model list the attributes

attributes

student
ID_number
first_name
surname
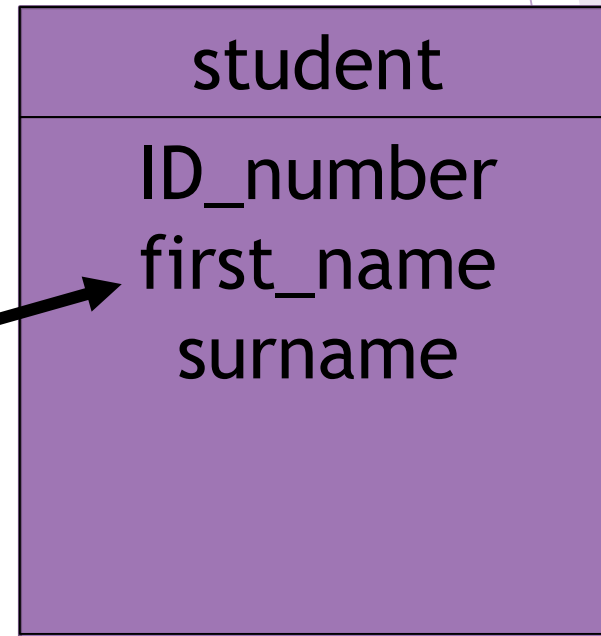
attributes are like column names

try not to use spaces

it is common to use an underscore (e.g. first_name)

the value can be:

▶ numeric

▶ string of characters

▶ date \ time

▶ other basic data types

▶ 'NULL' value (e.g. missing value)

    ▶ not the same as empty or zero

Database Design

table	columns	data types	notes
student	student_id	integer	primary key
student	st_firstname	varchar(50)	
lecturer	staff_id	integer	primary key
…	…		…
unit	student_id	integer	foreign key

wait

- we used id_number

- but what does this mean?

- is it easy to know what entity it is for?

- how about

                        student_id

we choose a primary key

or, create one

in the ER diagram we underline the primary key

primary key

student

<u>student_ID</u>
st_firstname
st_surname
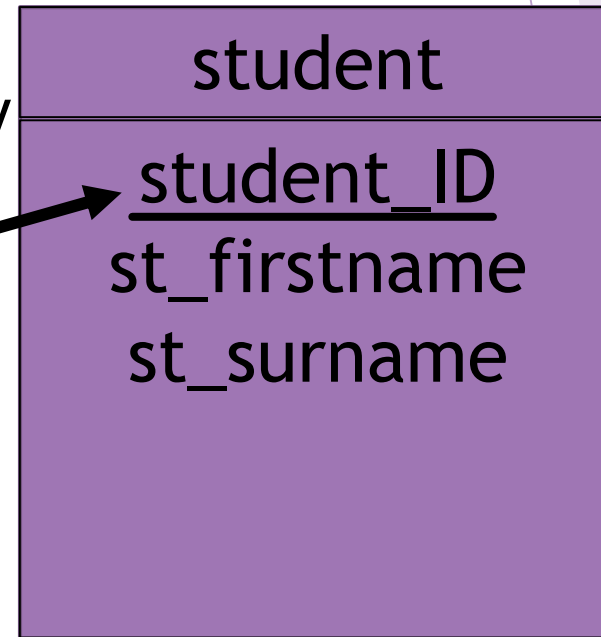
it is possible to use multiple attributes

but stick with the easy route for now

design & test data

does your design & test data match?

for example

- if you have name, does your test data only have one field for name?

- if you have first name and surname, does your test data have two fields for name?

test data

always save your test data

make versions in case you want to go back

everyone always makes mistakes – this is how we get better!

Relational Database Design

your task is to create an

Entity Relationship (ER) Diagram

- create entities, attributes and relationships
- this is an ER diagram

'Relational' database design
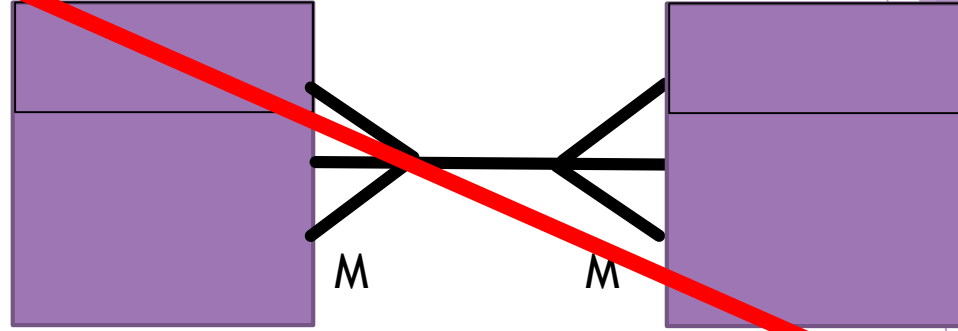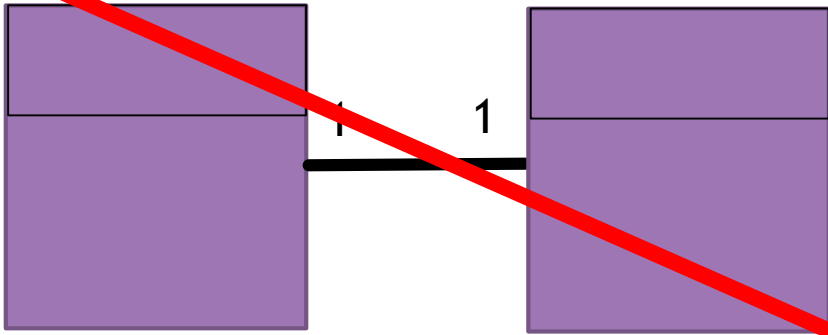
Relationship

association between entities

e.g.

- students take courses
- courses contain students

Two entities are related by the degree

- ▶ a course contains one or more students
- ▶ an instructor is assigned one and only one department
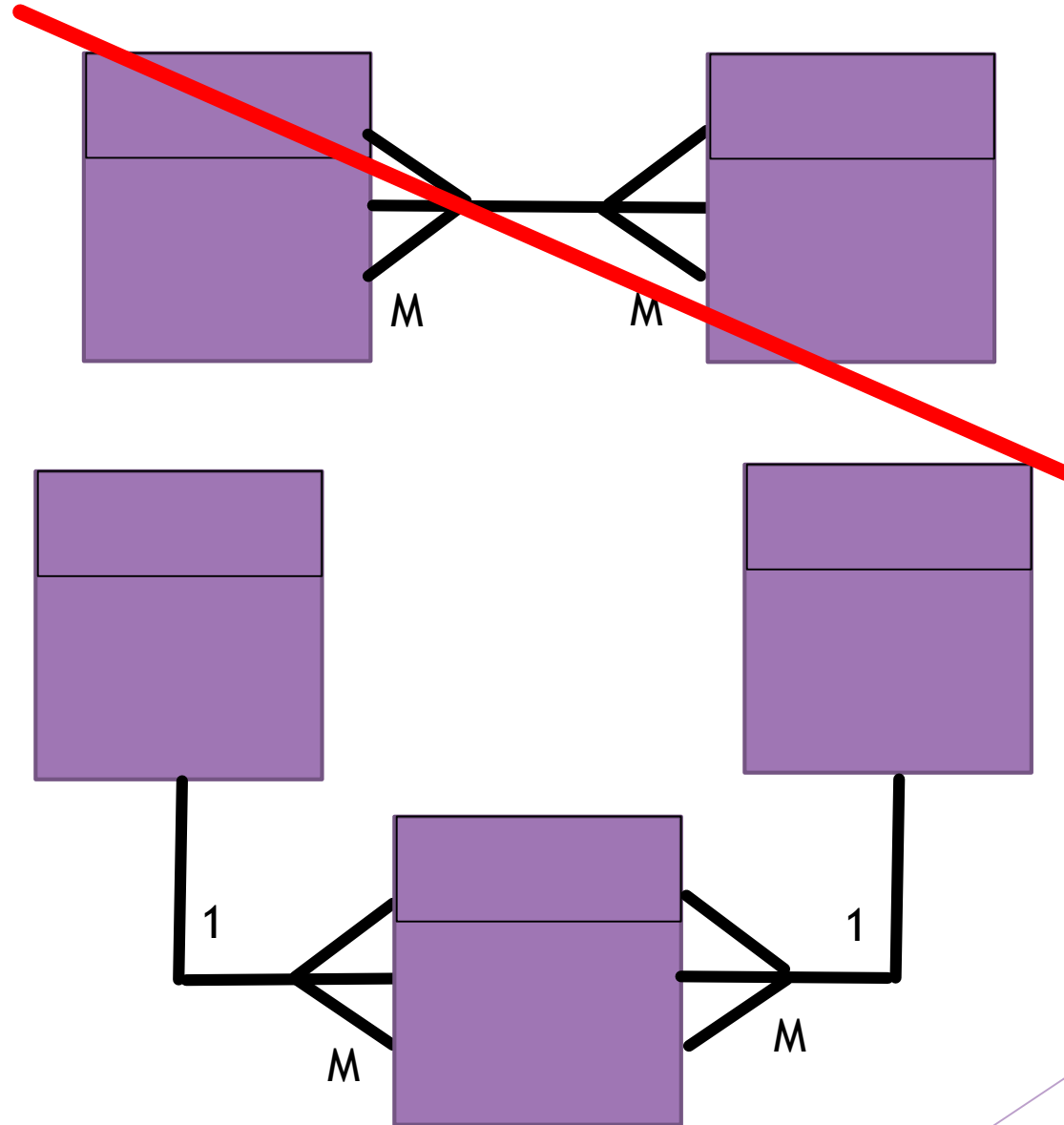
Relationships
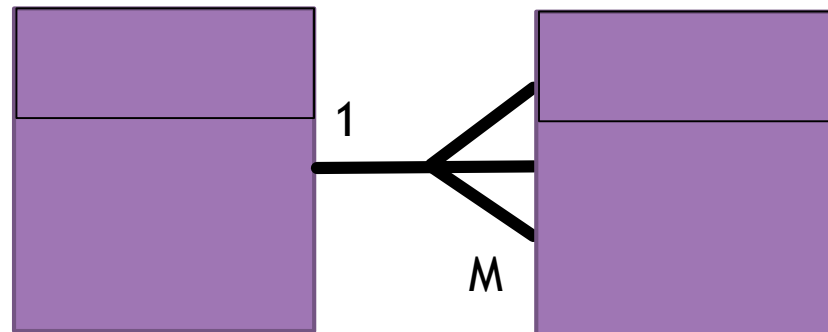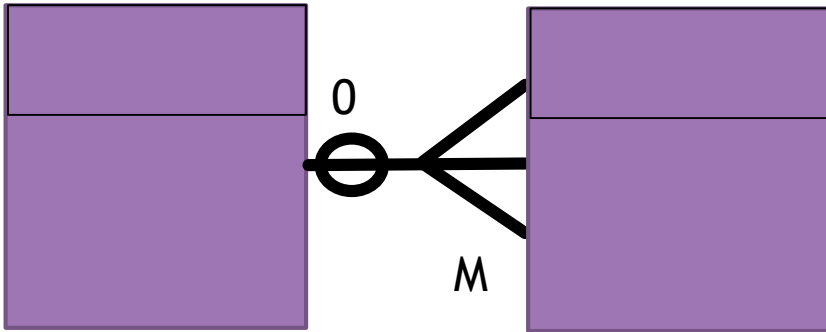
Relationship

- Look at the relationship both ways
- Each and every lecturer is assigned to one and only one department
- Each and every department contains one or many lecturers
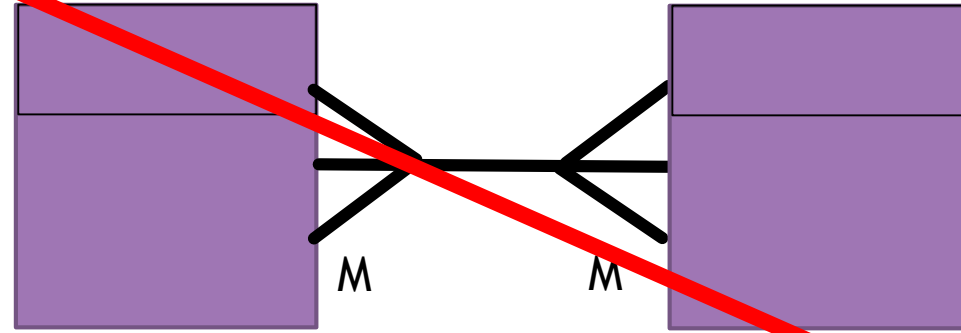- This is a 1 to many (1:M) relationship (this is good!)
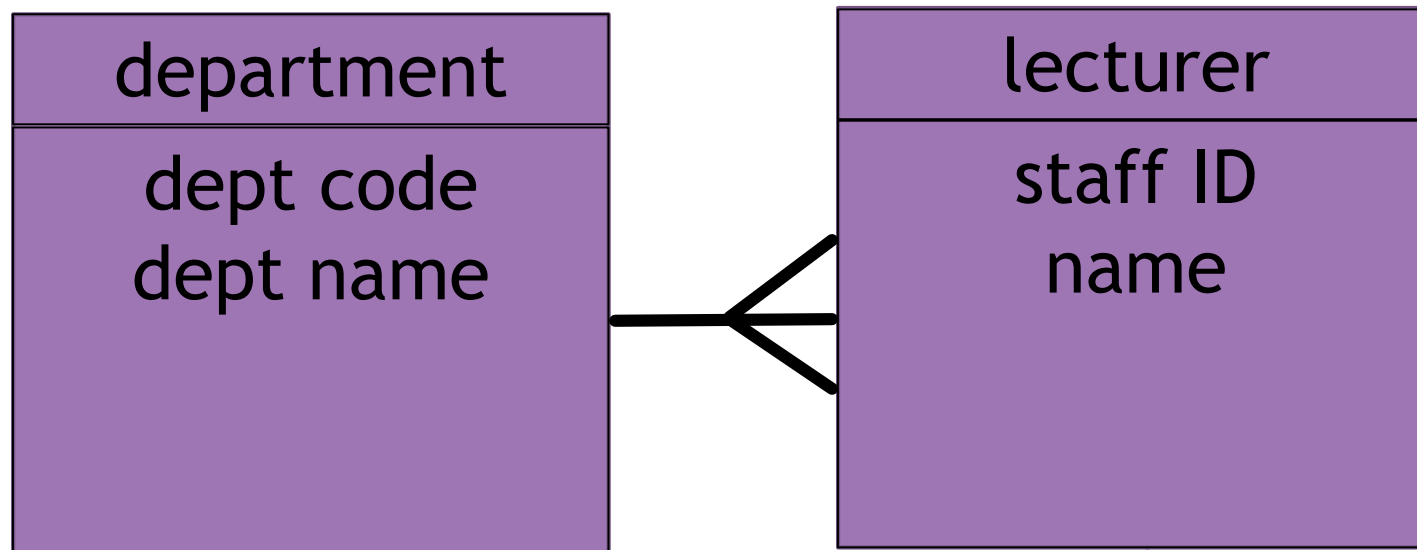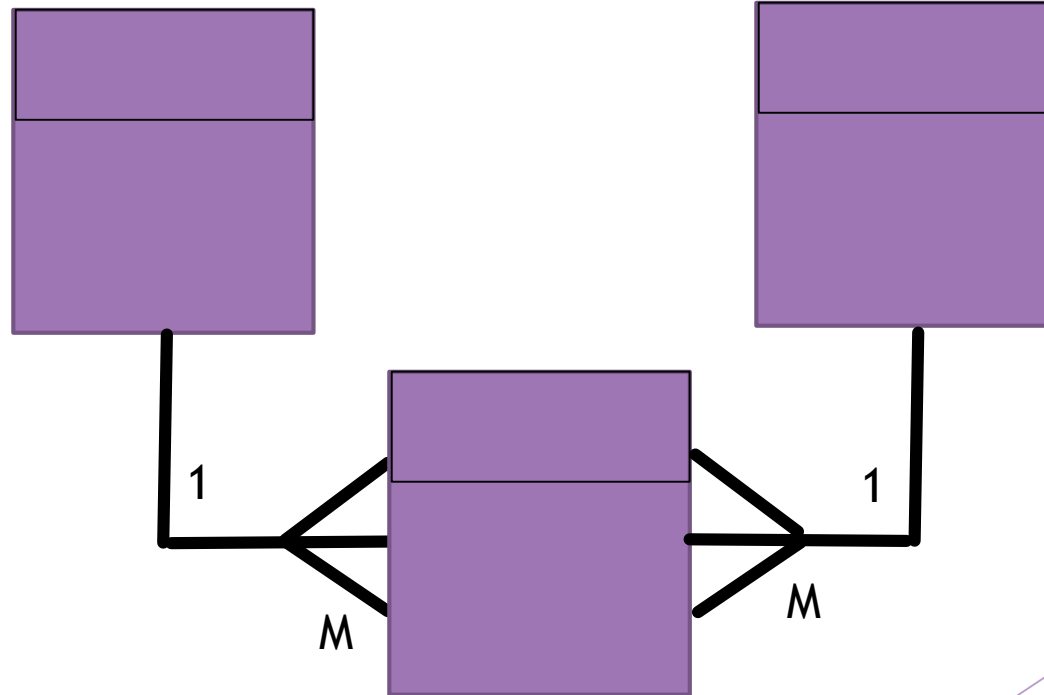
Relationships

Relationships

students — courses

M — M

students — courses

1 — 1

M — M

registration

identify and model entities   (tables)

identify and model attributes   (columns)

identify unique identifiers   (for each table)

identify and model relationships   (between tables)

resolve any design issues

- use junction tables rather than Many-to-Many

- if you have 1 to 1, then do you need it?

entities	->	tables
attributes	->	columns
unique identifiers	->	primary keys
relationships	->	foreign keys

Database Implementation

no spaces in names

▶ (SQL doesn't like them!)

keys – use INT type

▶ faster

▶ reduce errors, consistent format (10-digit phone number)

▶ can use CHAR for small/appropriate tables (e.g. CA,AL,AK)

text

▶ VARCHAR(50) preferred

Table and column names

► must start with a letter

► followed by letters, numbers, or underscores

► not to exceed a total of 30 characters in length

► not SQL reserved keywords

  ► e.g. "select", "create", "insert", etc.

Design your Database

your task is to create an

Entity Relationship (ER) Diagram

- create entities, attributes and relationships

- create your ER diagram
  - start with your tables / entities
  - start with your column names / attributes
  - underline primary keys
  - relationships
- also consider
  - data types (integer, date, varchar)
  - any restraints (set amount >= 0)

  you are allowed to use AI to create test data

create test data

to see if your data types are correct

save test data in excel

could you create an ER diagram
for a business,
like an online store ?

Thank you!

any questions?