# SOEN 363: Data Systems for Software Engineers

## Tutorial: Drawing ER Diagrams with draw.io (diagrams.net)

Parsa Kamalipour

Department of Computer Science and Software Engineering
Concordia University

Course Instructor: Hamed Jafarpour

Fall 2026

# Getting Started with draw.io

A tool for drawing ER diagrams

## Goal of this session

By the end of the tutorial you should be able to open draw.io, build a small ER diagram in the notation used in the lectures, and export it as a clean PDF or PNG.

**Next week's tutorial** covers ER modelling itself (entities, relationships, constraints). Today is about the tool.

# Outline

- Why a digital tool?
- What is draw.io and how to open it
- Interface tour
- ER notation used in this course and its draw.io equivalents
- Building a diagram step by step
  - Adding shapes
  - Connecting shapes
  - Styling: keys, thick lines, arrows, dashed boxes
  - Arranging and aligning
- Saving, exporting, and sharing
- Common mistakes
- Hands-on exercise

# Why a digital tool?

An ER diagram is a design document that other people read, discuss, and revise. A digital tool gives you:

- clean, consistent shapes and labels that are easy to read
- diagrams you can edit as the design changes, instead of redrawing from scratch
- exports that go straight into documents, slides, or a repository
- a file you can share and co-edit with others

Many tools can draw ER diagrams (Lucidchart, draw.io, Visio, . . . ). In the tutorials we use **draw.io** because it is:

- free and open source, no account required
- available in the browser and as a desktop app
- flexible enough to reproduce the exact notation from the lectures
- easy to share (plain text file, works with Git)

# What is draw.io?

- A general-purpose diagram editor, also known as **diagrams.net**
- **Browser version:** `https://app.diagrams.net` (nothing to install)
- **Desktop version:** Windows, macOS, Linux, from `https://www.drawio.com`
- **Integrations:** VS Code extension ("Draw.io Integration"), Google Drive, OneDrive, GitHub
- Diagrams are saved as `.drawio` files (XML text), so they can be versioned in Git like code
- Export formats: PNG, JPEG, SVG, PDF

## Recommendation

Use the browser version and save the `.drawio` file on your device (or in a Git repository).

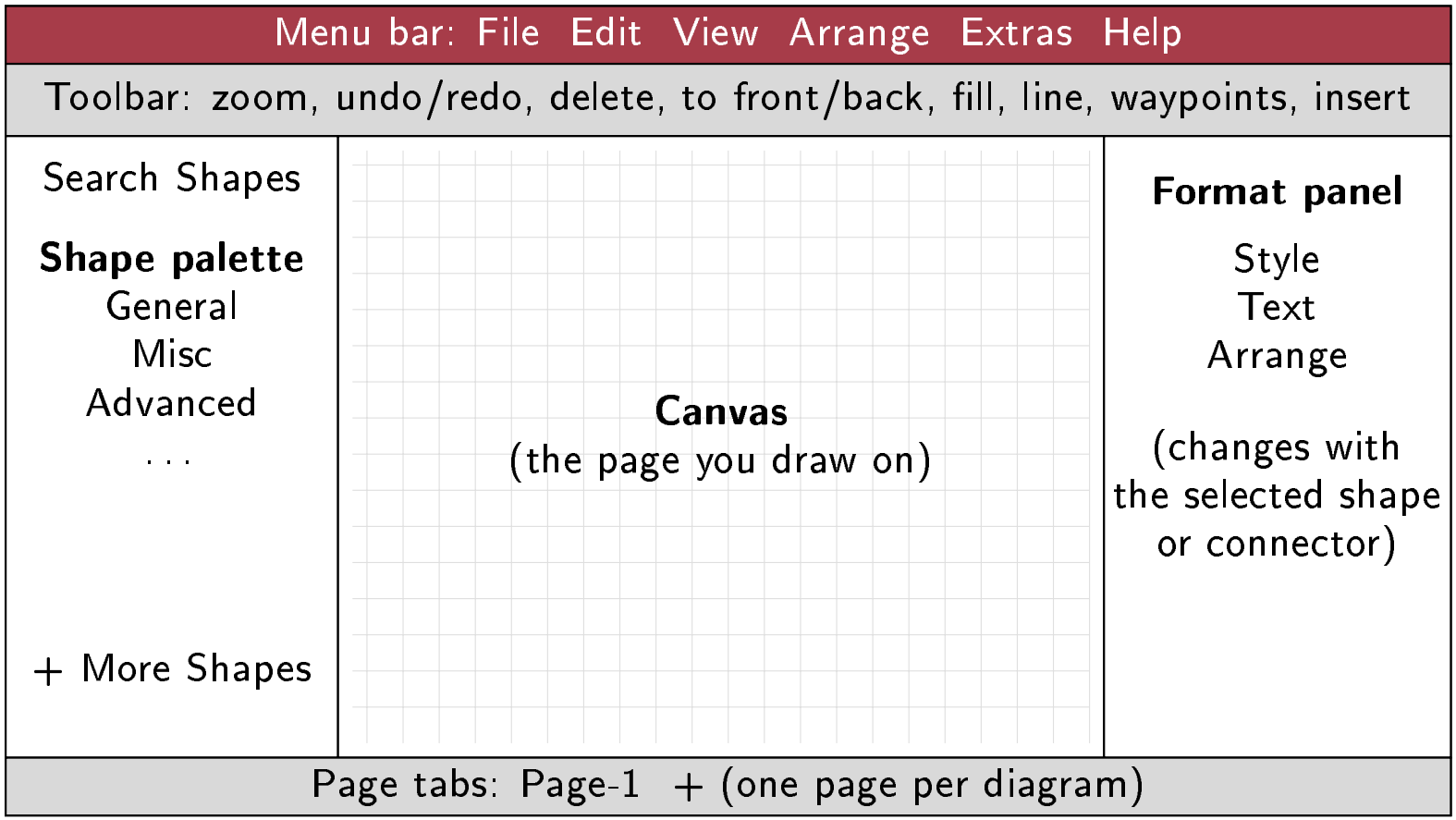
# Opening draw.io for the first time

1. Go to `https://app.diagrams.net`
2. Choose where to save your diagrams:
   - **Device**: saves a file on your computer (simplest, recommended)
   - **Google Drive** or **OneDrive**: lets several people edit the same file in real time
   - You can change this later under **File** → **Save as**
3. Click **Create New Diagram**
4. Select **Blank Diagram**, give it a name (e.g. `soen363_er.drawio`), click **Create**

**Note**

If you chose "Device", the browser may ask where to download the file the first time you save. Keep the `.drawio` extension.

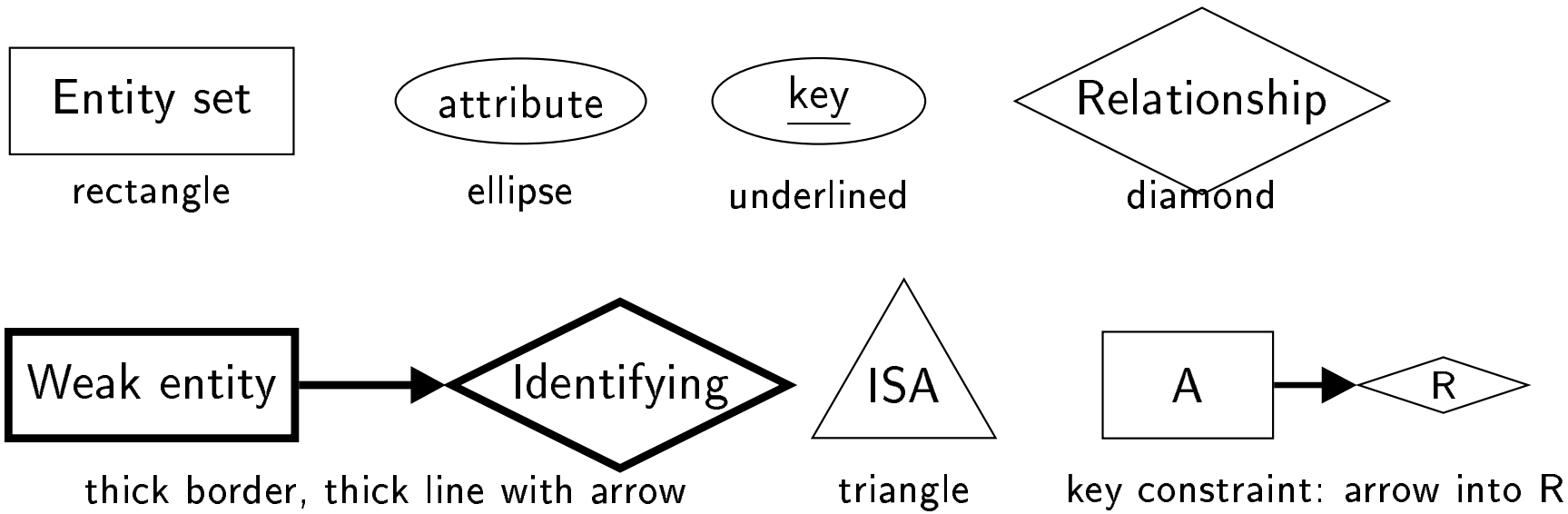
# The interface




- Hide or show the format panel with `Ctrl+Shift+P`; fit the diagram to the window with `Ctrl+Shift+H`; zoom with `Ctrl` + mouse wheel

# ER notation used in this course

Same notation as the lecture (Ramakrishnan & Gehrke):




- **Total participation**: thick line between entity set and relationship set
- **Aggregation**: dashed box around a relationship set and its entity sets
- Descriptive attributes of a relationship: ellipse attached to the diamond

# Course notation → draw.io

| ER concept | draw.io shape | Style setting |
|---|---|---|
| Entity set | General → Rectangle | default |
| Attribute | General → Ellipse | default |
| Primary key | text inside the ellipse | Text → Underline, `Ctrl+U` |
| Relationship set | General → Diamond | default |
| Plain ER line | connector | Style → Line end: *None* |
| Key constraint | connector | Style → Line end: *Classic* |
| Total participation | connector | Style → Line width: 3 pt |
| Weak entity set | Rectangle | Style → Line width: 3 pt |
| Identifying relationship | Diamond + connector | Line width 3 pt, arrow end |
| ISA | General → Triangle | type `ISA` inside |
| Aggregation | Rectangle, no fill | Pattern: *Dashed*; send to back |

## Important

draw.io also ships an "Entity Relation" library with table-style shapes and crow's-foot connectors. That is a different notation. Build your diagrams from the **General** shapes above.

# Step 1: Adding shapes

- **Click** a shape in the palette to drop it on the canvas, or **drag** it to a precise spot
- **Double-click** a shape (or select it and start typing) to set its label
- **Resize** with the corner handles; hold `Shift` to keep the aspect ratio
- **Duplicate** a styled shape with `Ctrl+D`, or `Ctrl` + drag
- **Search** the palette (top left) if you cannot find a shape, e.g. "diamond" or "triangle"

## Time saver

Style one ellipse the way you want (size, font, no fill), then select it and choose **Format** → **Style** → **Set as Default Style**. Every new ellipse will use that style. Do the same for rectangles and diamonds.

# Step 2: Connecting shapes

- Hover over a shape: blue arrows appear on its four sides
- **Drag** from a blue arrow onto another shape; release when the target shape shows a **blue outline**. This means the connector is *attached* and will follow the shape when you move it
- Connectors have an arrowhead by default. ER lines have none: select the connector, then **Format** → **Style** → **Line end** → **None**
- Fix all at once: **Edit** → **Select Edges** (`Ctrl+Shift+E`), then set Line end to None
- Straight vs. bent lines: **Format** → **Style** → **Waypoints** (Straight, Orthogonal, Curved)
- Drag the midpoint of a connector to add a bend; **Arrange** → **Waypoints** → **Clear Waypoints** to reset

## Check

Move a shape. If a line does not move with it, the line was never attached: delete it and reconnect.

# Step 3: Styling for the course notation

- **Underline a key**: double-click the ellipse, select the text, press `Ctrl+U`
- **Thick line** (total participation, weak entity, identifying relationship): select the object, **Format** → **Style** → **Line width**, set to 3 pt
- **Arrow** (key constraint): select the connector, **Line end** → **Classic**; make sure the arrow points *into the diamond*, not out of it (swap with **Arrange** → **Direction** → **Reverse** if needed)
- **Dashed box** (aggregation): add a rectangle, set **Fill** → **None** and **Pattern** → **Dashed**, resize it around the relationship and its entity sets, then **Arrange** → **To Back** (`Ctrl+Shift+B`)
- **Raw style** for fine control: `Ctrl+E` opens the style string, e.g. `ellipse;whiteSpace=wrap;html=1;strokeWidth=3;`

Keep fills white or none and use one font size for the whole diagram. Clean and readable beats colourful.

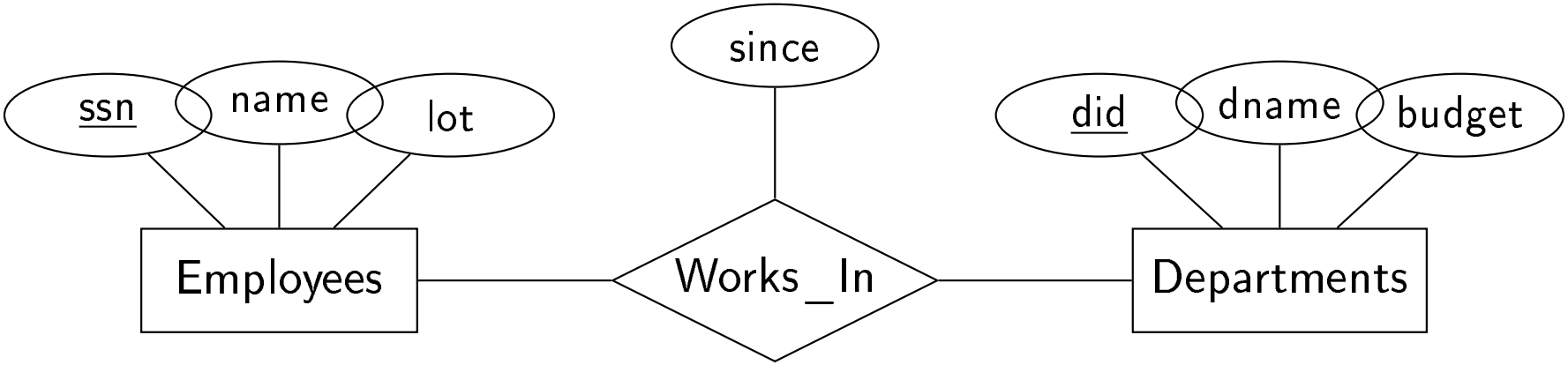
# Step 4: Arranging and aligning

- Select several shapes (drag a box around them, or `Shift` + click), then **Format** → **Arrange** → **Align** (left, centre, top, . . . ) and **Distribute** (equal spacing)
- Snap to grid is on by default; toggle the grid under **View** → **Grid**
- Nudge a selection one grid step with the arrow keys
- **Group** related shapes (an entity set with its attributes) with `Ctrl+G` so they move together; ungroup with `Ctrl+Shift+U`
- **Arrange** → **Layout** offers automatic layouts, but for ER diagrams manual placement usually looks better

## Layout tips

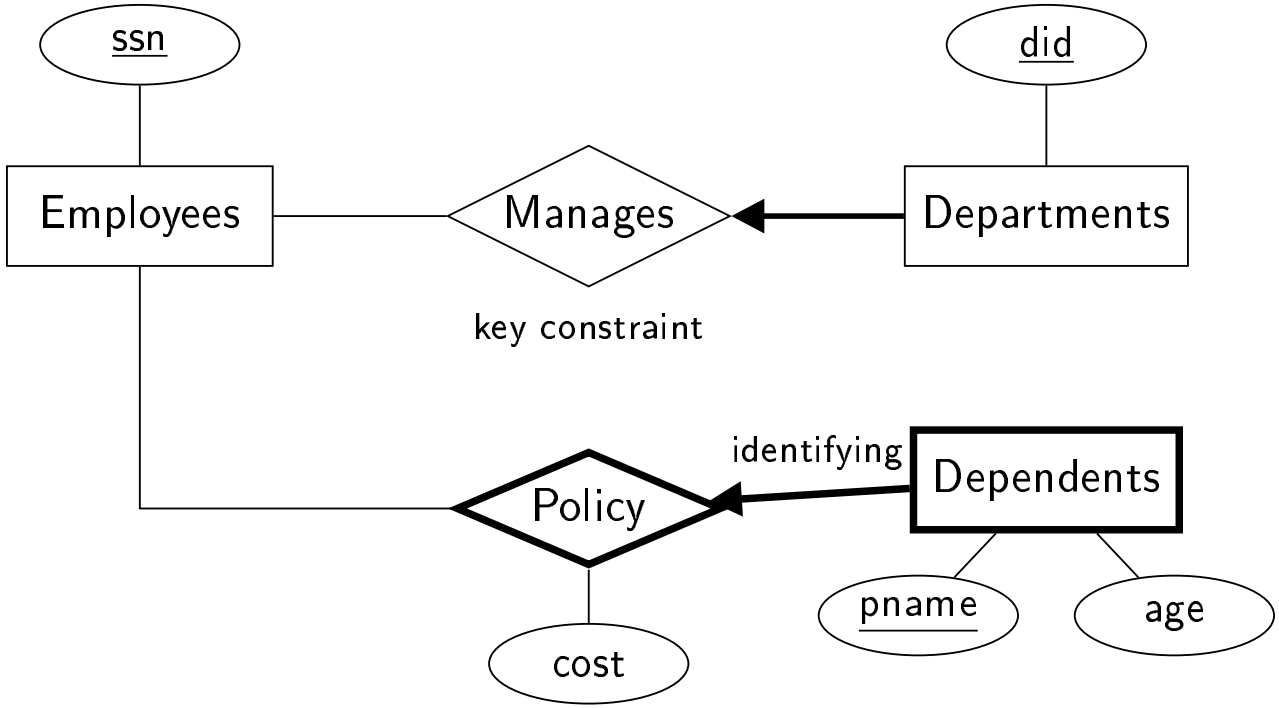
Put the central entity set (e.g. the one with the most relationships) in the middle, keep attributes close to their owner, and avoid crossing lines. Split very large diagrams across pages rather than shrinking the font.

# Worked example: the lecture's Works_In diagram




1. Add two rectangles (`Employees`, `Departments`) and one diamond (`Works_In`)
2. Add seven ellipses; underline `ssn` and `did` with `Ctrl+U`
3. Connect each ellipse to its owner, and each rectangle to the diamond; set Line end to None for all connectors (`Ctrl+Shift+E`, then Style)
4. Select the three ellipses above each rectangle and align them; `Ctrl+Shift+H` to fit the view

# Extending the example: constraints and weak entities




- `Departments` to `Manages`: Line end = Classic (at most one manager per department)
- `Dependents`, `Policy` and the connector between them: Line width = 3 pt, plus an arrow end on the connector
- No dashed underline in draw.io: underline the partial key `pname` normally and add a short note next to the diagram

# Saving and exporting

## Saving the editable file

- `Ctrl+S` or **File → Save** writes the `.drawio` file; keep it, this is the only editable version

## Exporting an image

- **File → Export as → PDF**: vector output, sharp at any zoom; best for LaTeX or Word documents
- **File → Export as → PNG**: set Zoom to 200 to 300%; tick *Include a copy of my diagram* so the PNG can be re-opened in draw.io
- **File → Export as → SVG**: vector, good for web or Google Docs
- Tick *Crop* to remove empty page space around the diagram

**Reminder**

An exported diagram must be readable at normal zoom. Never paste a blurry screenshot.

# Sharing a diagram

- **One source of truth**: a single `.drawio` file, either
  - committed to a Git repository (it is XML text, so it diffs and merges like code), or
  - stored on Google Drive / OneDrive and opened through draw.io, which supports real-time co-editing
- Agree on naming before you start: e.g. entity sets in `PascalCase`, attributes in `snake_case`, matching the table and column names you plan to use
- Use one page per diagram (bottom tabs); keep drafts on extra pages rather than in extra files
- If two people edit a Git-tracked file at the same time, expect merge conflicts. Take turns, or use the Drive option

## Diagram and schema

A relational schema should match its ER diagram. Keeping the diagram editable makes it easy to update both when the design changes.

# Common mistakes

1. **Floating connectors**: lines drawn near a shape but not attached. They drift when you move things. Always release on the blue outline
2. **Default arrowheads left on every line**: in this notation an arrow means a key constraint, so a stray arrowhead changes the meaning of your diagram
3. **Mixing notations**: crow's-foot or UML class boxes from the "Entity Relation" library next to Chen-style diamonds. Pick the lecture notation and use it everywhere
4. **Keys not underlined**: primary keys must be visibly marked
5. **Unreadable export**: tiny fonts, cropped edges, or a blurry PNG pasted into a document
6. **Losing the source file**: only the exported image survives, so the diagram cannot be edited later

# Hands-on exercise (about 20 minutes)

Recreate the diagrams from the lecture in draw.io:

1. `Employees`(ssn, name, lot) and `Departments`(did, dname, budget) with the many-to-many relationship `Works_In` and its descriptive attribute `since`
2. Add `Manages` between the same two entity sets with a key constraint on `Departments`
3. Add the weak entity set `Dependents`(pname, age) owned by `Employees` through the identifying relationship `Policy`(cost)
4. Add `Hourly_Emps`(hours_worked, hourly_wages) and `Contract_Emps`(contractid) under `Employees` using an `ISA` triangle
5. Export the result as PDF and open it to check that it is readable

## If you finish early

Add the lecture's aggregation example: `Projects`(pid, started_on, pbudget), `Sponsors`, and `Monitors`(until) from `Employees`, with a dashed box around `Sponsors` and its two entity sets.

# Useful shortcuts

| Shortcut | Action | Shortcut | Action |
|---|---|---|---|
| `Ctrl+S` | Save | `Ctrl+D` | Duplicate |
| `Ctrl+Z` / `Ctrl+Y` | Undo / Redo | `Ctrl+G` | Group |
| `Ctrl+A` | Select all | `Ctrl+Shift+U` | Ungroup |
| `Ctrl+Shift+E` | Select edges | `Ctrl+Shift+B` | To back |
| `Ctrl+Shift+I` | Select vertices | `Ctrl+Shift+F` | To front |
| `Ctrl+U` | Underline text | `Ctrl+E` | Edit style |
| `Ctrl+Shift+H` | Fit window | `Ctrl+Shift+P` | Toggle format panel |
| `Ctrl` + wheel | Zoom | `Delete` | Delete selection |

On macOS use `Cmd` instead of `Ctrl`. The full list is under **Help → Keyboard Shortcuts**.

# Resources and next steps

## Resources

- draw.io web app: `https://app.diagrams.net`
- Documentation: `https://www.drawio.com/doc/`
- Lecture slides: *Entity-Relationship (ER) Data Model* (Moodle)

## Before next tutorial

- Finish the hands-on exercise and keep your `.drawio` file
- Review the ER lecture slides: key constraints, participation, weak entities, ISA, aggregation

**Next tutorial:** ER modelling in practice, from a textual description to a complete diagram.