

# 111-1 Introduction to Database Systems -

## Final team project

### **Build an application with a database management system and AWS!**

In this semester, you've learned about database management systems. Now it's time to build an application and take advantage of it with your group members (**3~5 students in a group**).

The application can be for any purpose, such as (but not limited to):

- Outbreak near me
- Food around NYCU
- Course selection guide
- Stock price prediction or buying/selling notification
- Social network systems monitoring
- NBA or FIFA World Cup championship prediction
- ... and others

The application can be a web application, a pure console application, a LINE chatbot, or any form that can interact with users. For example, you can build a website that can automatically query the next movie within 10km (based on the system time and location), using an up-to-date movie database that can automatically be updated from any possible data sources (see next paragraph). Another example is building a website that can predict future stock prices, with a database that contains the historical data of stock prices, discussions from social network platforms, or any other data sources. Please use your creativity to build the application!

You might notice that you need “data” to establish the foundation of your application. You can use any method to gather the data you need, such as (but not limited to):

- Open data from the government  
For example, <https://data.gov.tw/>
  - with API or CSV files
- Web scraping  
For example,  
<https://www.freecodecamp.org/news/web-scraping-python-tutorial-how-to-scrape-data-from-a-website/>
- Social network systems API
- ... and others

The topics and data sources listed above are just examples. Please try to solve issues in your daily life!

## Final project requests:

The DBMS should be on the AWS service, but the application can be either on your local environment or AWS (you will receive bonus points if you build your application on AWS. **For example, a website that is hosted on AWS, or an API hosted on AWS**). You can use any programming language to build your application with DBMS, and you can use any DBMS, including PostgreSQL and others.

Please submit **a document in either Mandarin or English, codes for the application and/or data import, and a presentation and demo video in either Mandarin or English** (less than 10 mins, **including important contents from the document, and a demo of your application**) to the E3 system by **2023/01/05 23:59 (no late work will be accepted)**, the contents should include:

1. Motivations
2. Application description
3. Data sources and how you collect and import the data (manually or automatically)
4. Database schema (you can use the visualization tool in DBMS directly, and list constraints that are not in the figure)
5. The application's functions and the related SQL queries used for the function.

**The scoring criteria, as a group (Will be graded on completeness, correctness, and clarity, by the instructor&TAs 60%, and your peers 25%):**

1. Database design (30%)
  - a. It should be in "good" design
  - b. The completeness of constraints
  - c. Index
  - d. Normalization vs. performance trade-off, explain your choice
2. Integration between data and database (30%)
  - a. Database import methods and strategies
  - b. Database update strategies
3. Integration between the database and your application (20%)
  - a. Does the application really need a database?
  - b. Queries or related codes
4. Completeness of the document/video/introduction/application (10%)
5. Creativity and others (10%)

**The scoring criteria, as an individual (Will be graded on participation, by your group members 15%):**

1. Project contributions (5%)
2. Project participation (5%)
3. Easy to communicate/cooperate with as a team member (5%)

**Final project document format:**

1. **An introduction of your application, including why you want to develop the application and the main functions of your application.**

2. Database design - describe the schema of all your tables in the database, including keys and index, if applicable (why you need the keys, or why you think that adding an index is or is not helpful).
3. Database design - describe the normal form of all your tables. If the tables are not in BCNF, please include the reason for it (performance trade-off, etc.).
4. From the data sources to the database - describe the data source and the original format.
5. From the data sources to the database - describe the methods of importing the original data to your database and strategies for updating the data, if you have one.
6. Application with database - explain why your application needs a database.
7. Application with database - includes the queries that are performed by your application, how your application performed these queries (connections between application and database), and what is the cooperating functions for your application.
8. All the other details of your application that you want us to know.

## Q&A:

**Q: The file size of my video is large, can I upload it to the YouTube platform?**

**A:**

You can upload the video to Youtube, but please make sure that is publicly available with the share link and TAs and your peers can access it.

**Q: Can I use ORM to access the DB?**

**A:**

Of course, you can use any method to access the DB.

**Q: Can I upload my application to GitHub and submit the link as a reference?**

**A:**

You can upload your application to GitHub, but please make the repo public.

**Q: Can I use a publicly available framework to build my application?**

**A:**

Sure, you can use any framework to build the application.

## Important!

Please turn off (stop) the RDS service and End the Learner Lab when you are not working with the database to save money. You can always re-start the database afterward, and the data is still there.

Databases

Group resources

Modify

Actions

Restore from

Filter by databases

DB Identifier	Role	Engine	Region & AZ	Size	Status	CPU
database-1	Instance	PostgreSQL	us-east-1f	db.t3.micro	Available	

Start Lab

End Lab