

CS253: Software Development

Welcome to Lecture 11!

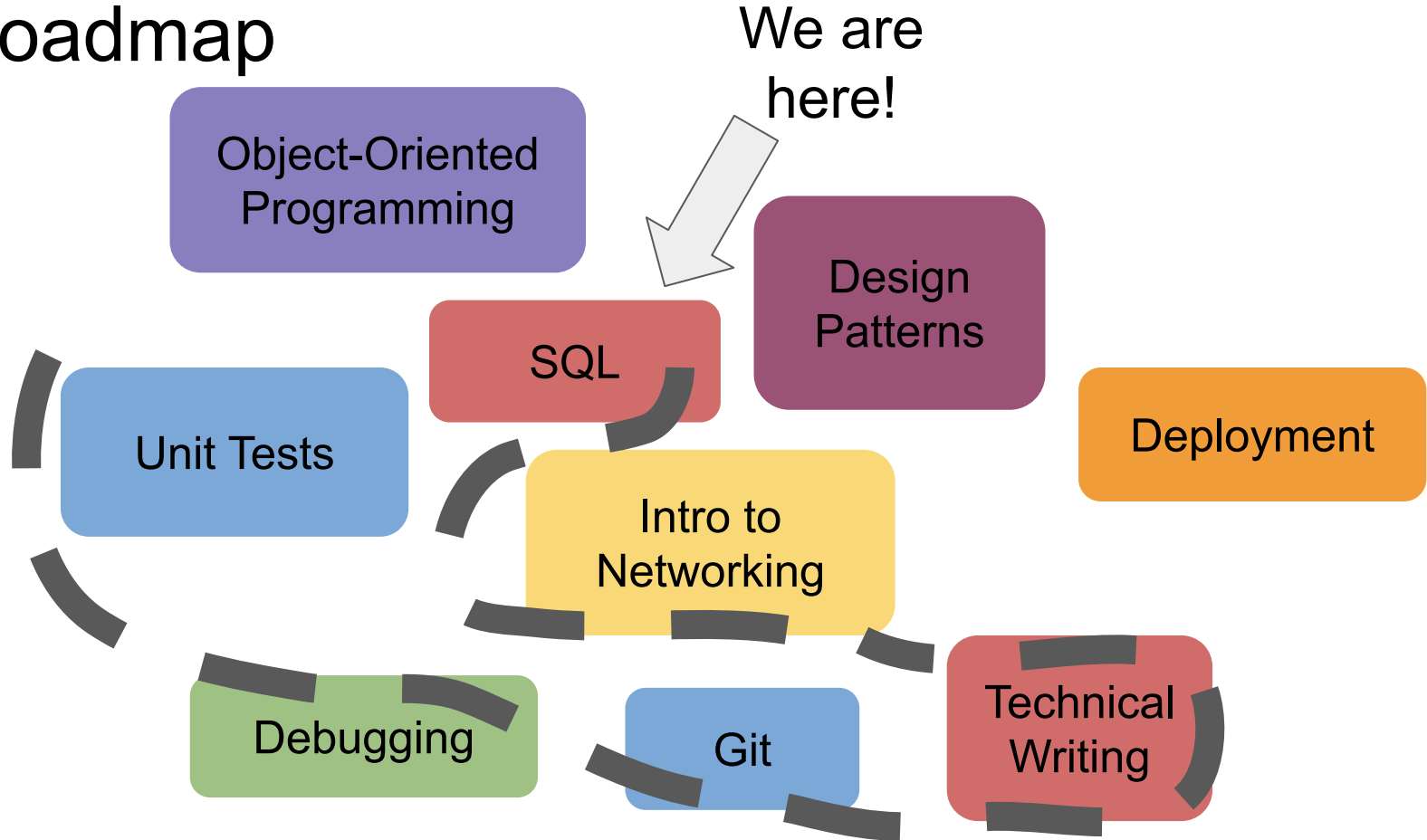
Daniel George

October 3, 2023

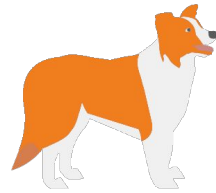
Announcements

- Today/this week: deep dive into SQL and data
- Assignment 2 is due on Friday 10/6, midnight CST
- Daniel is here on campus this week in CNS for lectures

Roadmap



SQL (Structured Query Language)



No logo! (See why on the next slide)

SQL

- SQL was not invented by a single individual; rather, a team of researchers including two important ones, Donald Chamberlin and Raymond Boyce
- SQL was introduced in a paper published in 1974 called "SEQUEL: A Structured English Query Language"
- It is the gold standard for managing and querying **relational databases**

Contents: The payment of grain and dates as temple stipends for twelve months.

	[Nisannu]	Atu	Sinshu	Duzu	Abu	Ululu	naphar	Tashritu	Arvshamma	Kislimu	Tebbitu	Shabatu	Adaru	naphar sk'um	naphar	a-wi-lu-tum	MU-BI-im			
							sha i-na qit mIfu	nabi	mak-	rum				sha qit libbi aballi i-na libbi she'i sha Zarat- IMki u subuppu mahrum						
	72qa	72qa	72qa	72qa	72qa	72qa	2gur	72qa	72qa	72qa	72qa	72qa	72qa	2gur	72qa	4gur	144qa	KAL	mSin-da-ma-gu, "overseer."	
	72qa	72qa	72qa	72qa	72qa	72qa	2gur	72qa	72qa	72qa	72qa	72qa	72qa	2gur	72qa	4gur	144qa	KAL	mIp-pa-c-a ù(ditto)	
5	72qa	72qa	72qa	72qa	72qa	72qa	2gur	72qa	72qa	72qa	72qa	72qa	72qa	2gur	72qa	4gur	144	KAL	mIdinanni-Shamash, "keeper."	
	30qa	30qa	30qa	30qa	30qa	30qa	1 gur	30qa	30qa	30qa	30qa	30qa	30qa	1 gur	2 gur			SAL or i	Tam-bi-Da-du, "his wife."	
	24qa	24qa	24qa	24qa	24qa	24qa	144 qa	24qa	24qa	24qa	24qa	24qa	24qa	144 gur	1gur	108qa		SAL-TUR	Da-ti-lu-sha, "his daughter", "seeress."	
	18qa	18qa	18qa	18qa	18qa	18qa	108 qa											KAL-TUR	mArdu-Nusku, "his son," ultu Tashritu harranu.	
	12qa	12qa	12qa	12qa	12qa	12qa	72 qa	12qa	12qa	12qa	12qa	12qa	12qa	72 qa	144 qa			KAL-TUR-TUR	mNusku-ki-na-u-gur, "his [grand] son."	
10	6qa	6qa	6qa	6qa	6qa	6qa	36 qa	6qa	6qa	6qa	6qa	6qa	6qa	36 qa	72 qa			TUR-GAB	mGab-mar-ta-ash, "his son."	
	72qa	72qa	72qa	72qa	72qa	72qa	2gur	72qa	72qa	72qa	72qa	72qa	72qa	2gur	72qa	4gur	144qa	KAL	mA-na-šShe-mi-i-at-kul, "grinder."	
	48qa	48qa	48qa	48qa	48qa	48qa	1gur	108qa	48qa	48qa	48qa	48qa	48qa	1gur	108qa	3gur	36qa		SAL or i	Ish-tar-be-li-uš-ri.
																		KAL-TUR	mUshab-shi-uz-ni-a-na-ili, "her son," harranu.	
	24qa	24qa	24qa	24qa	24qa	24qa	144 qa	24qa	24qa	24qa	24qa	24qa	24qa	144 qa	1gur	108qa		KAL-TUR-TUR	mDu-ak-ki-in-itu, *SHI(?) "her [grand] son."	
15	30qa	30qa	30qa	30qa	30qa	30qa	1 gur	30qa	30qa	30qa	30qa	30qa	30qa	1 gur	2 gur			SAL-TUR	Ba-su-un-du, "her daughter," "seeress."	
	12qa	12qa	12qa	12qa	12qa	12qa	72 qa	12qa	12qa	12qa	12qa	12qa	12qa	72 qa	144 qa			SAL-TUR-GAB	Hu-la-la-tum, "her daughter."	
	6qa	6qa	6qa	6qa	6qa	6qa	36 qa	6qa	6qa	6qa	6qa	6qa	6qa	36qa	72 qa			TUR-GAB	I-na-rish-Marduk-di-nu, "her son."	
	48qa	48qa	48qa	48qa	48qa	48qa	1gur	108qa	48qa	48qa	48qa	48qa	48qa	1gur	108qa	3gur	36qa		SAL or i	Bi'il-balafu-t'rish(-ish)
	18qa	18qa	18qa	18qa	18qa	18qa	108 qa	18qa	18qa	18qa	18qa	18qa	18qa	108 qa	1gur	36qa		KAL-TUR-TUR	mLul-ta-mar-Nusku, "her son," "weaver."	
20	12qa	12qa	12qa	12qa	12qa	12qa	72 qa	12qa	12qa	12qa	12qa	12qa	12qa	72 qa	144 qa			SAL-TUR-GAB	Rabī-sha-šIsh-ha-ra, "her daughter."	
	6qa	6qa	6qa	6qa	6qa	6qa	36 qa	6qa	6qa	6qa	6qa	6qa	6qa	36 qa	72 qa			SAL-TUR-GAB	Di-ni-ili-lu-maw, "her daughter."	
	48qa	48qa	48qa	48qa	48qa	48qa	1gur	108qa	48qa	48qa	48qa	48qa	48qa	1gur	108qa	3gur	36qa		SAL or i	Mi-sha-ri-tum.
	48qa	48qa	48qa	48qa	48qa	48qa	1gur	108qa	48qa	48qa	48qa	48qa	48qa	1gur	108qa	3gur	36qa		SAL or i	I-na-Ak-ka-di-rab-bal.

Databases are Old!

- Based on what we see in the diagram on the previous slide, we can conclude that:
 - A **table** stores some set of information (here, worker stipends)
 - Every **row** in a table stores one item in that set (here, one worker)
 - Every **column** has some attribute of that item (here, the stipend for a particular month)

Why Even Use Databases?

- Three reasons to use database instead of just spreadsheets, like Excel or Google Sheets:
 - **Scale:** Databases can store not just items numbering to tens of thousands but even millions and billions
 - **Update Capacity:** Databases are able to handle multiple updates of data in a second
 - **Speed:** Databases allow faster look-up of information. This is because databases provide us with access to different algorithms to retrieve information. In contrast, spreadsheets that merely allow the use of Ctrl+F or Cmd+F to go through hits one at a time

So, what is a Database

- A database is a **way of organizing data such that you can perform four operations on it:**
 - Create
 - Read
 - Update
 - Delete
- A **database management system (DBMS)** is a way to interact with a database using a graphical interface or textual language.
- Examples of DBMS: MySQL, Oracle, PostgreSQL, SQLite (we saw last week), Microsoft Access, MongoDB etc.

SELECT

- What data is actually *in* our database?
- Couple of query examples

```
SELECT *  
FROM "table";
```

```
SELECT "title"  
FROM "table";
```

```
SELECT "title", "author"  
FROM "table";
```

LIMIT

- What if we have millions of rows, can we select just some?

```
SELECT "title"  
FROM "table"  
LIMIT 10;
```

WHERE

- We can select certain rows based on a condition

```
SELECT "title"  
FROM "table"  
WHERE "year" = 2023;
```

- Operators: =, !=, <>

WHERE

- Keywords: AND, NOT, OR

```
SELECT "title"  
FROM "table"  
WHERE ("year" = 2023 OR "year" = 2022)  
AND "format" != 'hardcover';
```

NULL

- It is possible to have missing data

```
SELECT "title"  
FROM "table"  
WHERE "author" IS NOT NULL;
```

LIKE

- We can select data that matches a specified string
- **LIKE** is combined with the operators % (matches any characters around a given string) and _ (matches a single character)

```
SELECT "title"  
FROM "table"  
WHERE "title" LIKE '%mockingbird%';
```

Ranges

- We can use different operators `<`, `>`, `<=`, `>=` to select data based on a range of values

```
SELECT "title"  
FROM "table"  
WHERE "year" >= 2019 AND "year" <  
2023;
```


ORDER BY

- We can return data in some specified order

```
SELECT "title", "rating"  
FROM "table"  
ORDER BY "rating";
```

```
SELECT "title", "rating"  
FROM "table"  
ORDER BY "rating" DESC LIMIT 10;
```

CREATE TABLE

- How to make a table

```
CREATE TABLE books (  
    "id",  
    "author",  
    "title",  
    "rating",  
    "num_pages",  
    "year"  
);
```

INSERT INTO

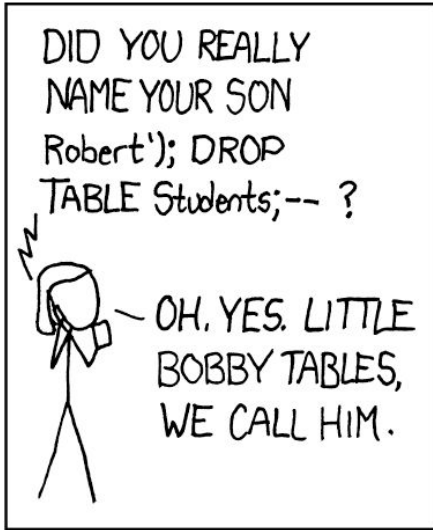
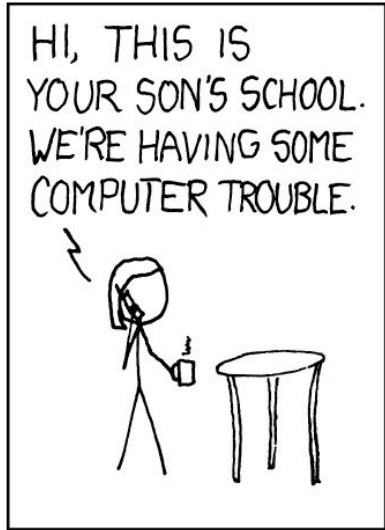
- How to add records to a table

```
INSERT INTO "books" ("id", "author",  
    "title", "rating", "num_pages", "year")  
VALUES (1, 'Harper Lee', 'To Kill A  
    Mockingbird', 5, 384, '1-1-1960');
```

DROP TABLE

- How to delete (drop) a table

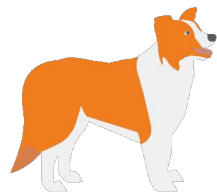
```
DROP TABLE "books" ;
```





Practice

What are your questions?





Thank you!