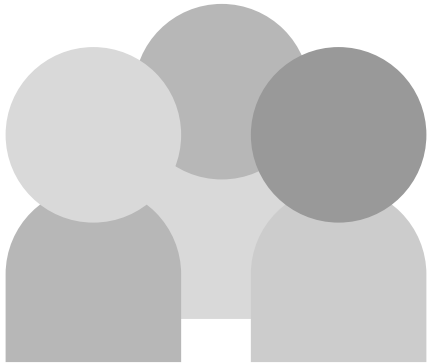
This is CS50



roster.db

roster.db

id	student_name	house	head
1	Adelaide Murton	Slytherin	Severus Snape
2	Adrian Pucey	Slytherin	Severus Snape
3	Anthony Goldstein	Ravenclaw	Filius Flitwick
…	…	…	…

roster.db

students

id	student_name	house	head
1	Adelaide Murton	Slytherin	Severus Snape
2	Adrian Pucey	Slytherin	Severus Snape
3	Anthony Goldstein	Ravenclaw	Filius Flitwick
…	…	…	…

Selecting

```
SELECT *
FROM students;
```

roster.db

students

id	student_name	house	head
1	Adelaide Murton	Slytherin	Severus Snape
2	Adrian Pucey	Slytherin	Severus Snape
3	Anthony Goldstein	Ravenclaw	Filius Flitwick
…	…	…	…

```
SELECT student_name
FROM students;
```

roster.db

students

id	student_name	house	head
1	Adelaide Murton	Slytherin	Severus Snape
2	Adrian Pucey	Slytherin	Severus Snape
3	Anthony Goldstein	Ravenclaw	Filius Flitwick
…	…	…	…

```sql
SELECT student_name
FROM students
WHERE house = 'Slytherin';
```

roster.db

students

id	student_name	house	head
1	Adelaide Murton	Slytherin	Severus Snape
2	Adrian Pucey	Slytherin	Severus Snape
3	Anthony Goldstein	Ravenclaw	Filius Flitwick
…	…	…	…

```
SELECT student_name
FROM students
WHERE student_name LIKE "%Potter";
```

Find the names and houses of students whose names begin with "H".

Ordering

```
SELECT *
FROM students;
```

```
SELECT *
FROM students
ORDER BY student_name;
```

```sql
SELECT *
FROM students
ORDER BY student_name ASC;
```

```sql
SELECT *
FROM students
ORDER BY student_name DESC;
```

Sort the student data in alphabetical order, first by house and then by student name.

Limiting

```
...
LIMIT 1;
```

Aggregating

```
SELECT COUNT(*)
FROM students;
```

Count the number of students in Gryffindor

Count the number of students in each house

```
SELECT house, COUNT(*)
FROM students
GROUP BY house;
```

Database Design

Design Principles

- Each table should be a collection of a **single entity**.
- For example, we should have a different table for each of **students**, **houses**, and **student-house assignments**.

roster.db

students

id	student_name	house	head
1	Adelaide Murton	Slytherin	Severus Snape
2	Adrian Pucey	Slytherin	Severus Snape
3	Anthony Goldstein	Ravenclaw	Filius Flitwick
…	…	…	…

roster.db

students

id	student_name
1	Adelaide Murton
2	Adrian Pucey
3	Anthony Goldstein
…	…

roster.db

students

id	student_name
1	Adelaide Murton
2	Adrian Pucey
3	Anthony Goldstein
…	…

houses

id	house	head
1	Gryffindor	Minerva McGonagall
2	Hufflepuff	Pomona Sprout
3	Ravenclaw	Filius Flitwick
4	Slytherin	Severus Snape

Design Principles

- Each piece of data should be stored in a **single location**, and thereafter referred to by its **ID** ("primary key").
- For example, we should ensure every student and house has an ID, then use those IDs in the house assignments table.

roster.db

students

id	student_name
1	Adelaide Murton
2	Adrian Pucey
3	Anthony Goldstein
…	…

houses

id	house	head
1	Gryffindor	Minerva McGonagall
2	Hufflepuff	Pomona Sprout
3	Ravenclaw	Filius Flitwick
4	Slytherin	Severus Snape

assignments

id	student_id	house_id
1	1	4
…	…	…

```
CREATE TABLE houses (
    id,
    house,
    head,
    PRIMARY KEY(id)
);
```

```
CREATE TABLE houses (
    id INTEGER,
    house TEXT,
    head TEXT,
    PRIMARY KEY(id)
);
```

```
CREATE TABLE houses (
    id INTEGER NOT NULL,
    house TEXT NOT NULL,
    head TEXT NOT NULL,
    PRIMARY KEY(id)
);
```

NOT NULL

"required"

roster.db

students

id	student_name	house	head
1	Adelaide Murton	Slytherin	Severus Snape
2	Adrian Pucey	Slytherin	Severus Snape
3	Anthony Goldstein	Raven claw	Filius Flitwick
…	…	…	…

houses

id	house	head

```
INSERT INTO tablename (column1, column2)
VALUES (value1, value2);
```

```
INSERT INTO houses (house, head)
VALUES ('Gryffindor', 'McGonagall');
```

roster.db

students

id	student_name	house	head
1	Adelaide Murton	Slytherin	Severus Snape
2	Adrian Pucey	Slytherin	Severus Snape
3	Anthony Goldstein	Raven claw	Filius Flitwick
…	…	…	…

houses

id	house	head
1	Gryffindor	McGonagall

roster.db

students

id	student_name
1	Adelaide Murton
2	Adrian Pucey
3	Anthony Goldstein
…	…

houses

id	student_name	head
1	Gryffindor	Minerva McGonagall
2	Hufflepuff	Pomona Sprout
3	Ravenclaw	Filius Flitwick
4	Slytherin	Severus Snape

assignments

id	student_id	house_id
1	1	4
…	…	…

Count the number of students in Gryffindor

SELECTs
JOINs

SELECTs

JOINs

roster.db

students

id	student_name
1	Adelaide Murton
2	Adrian Pucey
3	Anthony Goldstein
…	…

houses

id	student_name	head
1	Gryffindor	Minerva McGonagall
2	Hufflepuff	Pomona Sprout
3	Ravenclaw	Filius Flitwick
4	Slytherin	Severus Snape

assignments

id	student_id	house_id
1	1	4
…	…	…

roster.db

students

id	student_name
1	Adelaide Murton
2	Adrian Pucey
3	Anthony Goldstein
…	…

houses

id	student_name	head
1	Gryffindor	Minerva McGonagall
2	Hufflepuff	Pomona Sprout
3	Ravenclaw	Filius Flitwick
4	Slytherin	Severus Snape

assignments

id	student_id	house_id
1	1	4
…	…	…

```
SELECT id
FROM houses
WHERE house = 'Gryffindor'
```

roster.db

students

id	student_name
1	Adelaide Murton
2	Adrian Pucey
3	Anthony Goldstein
…	…

houses

id	student_name	head
1	Gryffindor	Minerva McGonagall
2	Hufflepuff	Pomona Sprout
3	Ravenclaw	Filius Flitwick
4	Slytherin	Severus Snape

assignments

id	student_id	house_id
1	1	4
…	…	…

```
SELECT COUNT(student_id)
FROM assignments
WHERE house_id =
(
    SELECT id
    FROM houses
    WHERE house = 'Gryffindor'
);
```

```
SELECT COUNT(student_id)
FROM assignments
WHERE house_id =
(
    1
);
```

SELECTs

JOINs

Count the number of students in Gryffindor

roster.db

students

id	student_name
1	Adelaide Murton
2	Adrian Pucey
3	Anthony Goldstein
…	…

houses

id	student_name	head
1	Gryffindor	Minerva McGonagall
2	Hufflepuff	Pomona Sprout
3	Ravenclaw	Filius Flitwick
4	Slytherin	Severus Snape

assignments

id	student_id	house_id
1	1	4
…	…	…

roster.db

students

id	student_name
1	Adelaide Murton
2	Adrian Pucey
3	Anthony Goldstein
…	…

houses

id	student_name	head
1	Gryffindor	Minerva McGonagall
2	Hufflepuff	Pomona Sprout
3	Ravenclaw	Filius Flitwick
4	Slytherin	Severus Snape

assignments

id	student_id	house_id
1	1	4
…	…	…

assignments JOIN houses

assignments houses

student_id	house_id	id	house	head
1	1	1	Slytherin	Severus Snape
2	1	1	Slytherin	Severus Snape
3	2	2	Ravenclaw	Filius Flitwick

```sql
SELECT *
FROM assignments
JOIN houses
ON assignments.house_id = houses.id;
```

```sql
SELECT *
FROM assignments
JOIN houses
ON assignments.house_id = houses.id;
```

```
SELECT COUNT(student_id)
FROM assignments
JOIN houses
ON assignments.house_id = houses.id
WHERE house = 'Gryffindor';
```

This was CS50