STRUCTURED QUERY LANGUAGE

W4D1

DATA PERSISTENCE

DATA PERSISTENCE

CSV
JSON
XML

RELATIONAL DATABASES

Reduce the **duplication** of data.
Provide a **robust query language.**
Accurately **model reality.**

DESIGN

>

DATABASE DENORMALIZATION

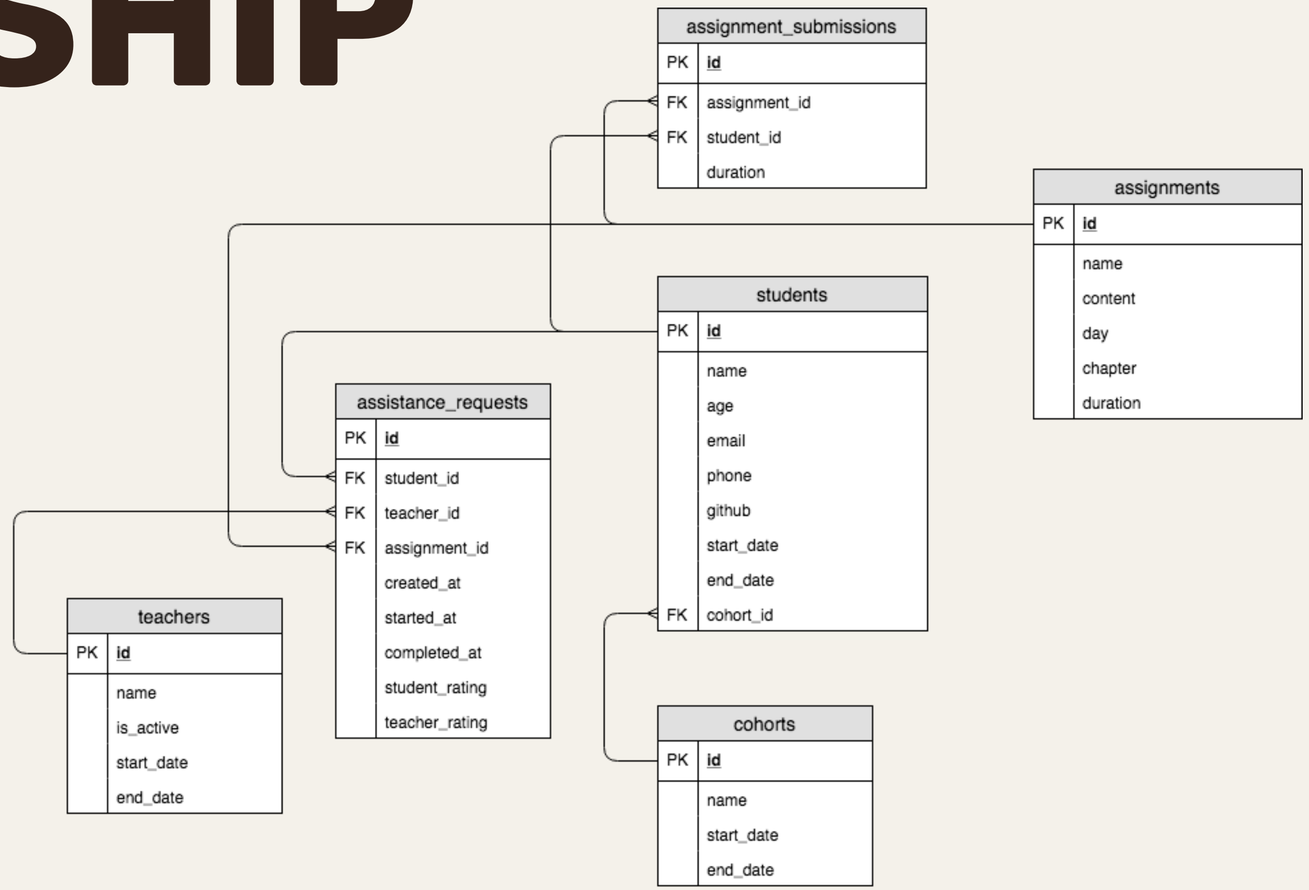
students		
id	name	cohort_name
1	Sam Billings	FEB12
2	Susan Hudson	MAR12
3	Malloy Jenkins	**APR09**
4	Maximilian Alesio	**APR09**
5	Pegasus Larue	**APR09**

DATABASE NORMALIZATION

students		
id	name	cohort_id
1	Sam Billings	1
2	Susan Hudson	2
3	Malloy Jenkins	**3**
4	Maximilian Alesio	**3**
5	Pegasus Larue	**3**

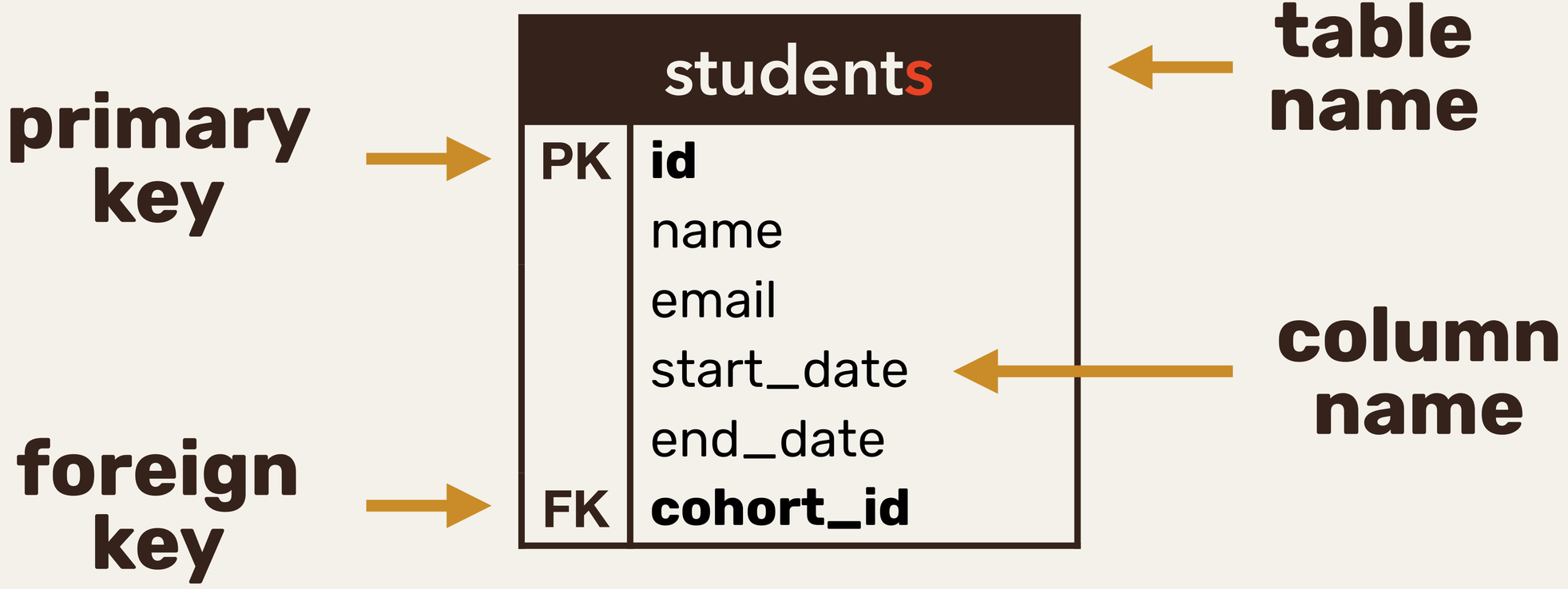
cohorts	
id	name
1	FEB12
2	MAR12
3	APR09

ENTITY RELATIONSHIP DIAGRAM



STICK TO CONVENTION!

STICK TO CONVENTION!



THE **FOREIGN KEY** IS ON THE **MANY SIDE**.

THE **FOREIGN KEY** IS ON THE **MANY SIDE**.

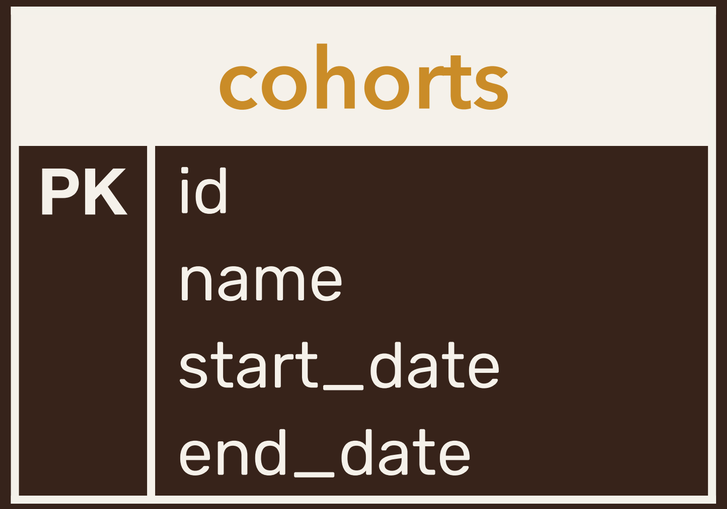
JOIN TABLES

assistance_requests	
PK	id
FK	student_id
FK	teacher_id
FK	assignment_id
	created_at
	started_at
	completed_at
	student_rating
	teacher_rating

SCHEMA

>

DATA TYPES



INTEGER
CHAR
VARCHAR
DATE
TEXT
BOOLEAN
SMALLINT
BIGINT
SERIAL

SMALLINT

+/- 32,767

INTEGER

+/- 2,147,483,647

BIGINT

+/- 9,223,372,036,854,775,807

ISO 8601

2018-04-26T21:36:28+00:00

800 555 4567

800-555-4567

(800)-555-4567

+1 800 555 4567

+62 78 402 55 45

8,005,554,567

INDEXING

```
const students = [
  { id: “cjvbZq”, name: “Tajana” },
  { id: “DJHa7I”, name: “Delfina” },
  { id: “hEWmRa”, name: “Filip” },
  { id: “Ck9YNg”, name: “Amie” },
  { id: “AG7piI”, name: “Radoslav” },
  { id: “rtwj0V”, name: “Embla” },
  { id: “VbnpoA”, name: “Natalia” }
]
```

```
const students = {
  “cjvbZq”: { id: “cjvbZq”, name: “Tajana” },
  “DJHa7I”: { id: “DJHa7I”, name: “Delfina” },
  “hEWmRa”: { id: “hEWmRa”, name: “Filip” },
  “Ck9YNg”: { id: “Ck9YNg”, name: “Amie },
  “AG7piI”: { id: “AG7piI”, name: “Radoslav” },
  “rtwj0V”: { id: “rtwj0V”, name: “Embla” },
  “VbnpoA”: { id: “VbnpoA”, name: “Natalia” }
}
```

INDEXING

```
const students = [
  { id: “cjvbZq”, name: “Tajana” },
  { id: “DJHa7I”, name: “Delfina” },
  { id: “hEWmRa”, name: “Filip” },
  { id: “Ck9YNg”, name: “Amie” },
  { id: “AG7piI”, name: “Radoslav” },
  { id: “rtwj0V”, name: “Embla” },
  { id: “VbnpoA”, name: “Natalia” }
]
```

```
const students = {
  “cjvbZq”: { id: “cjvbZq”, name: “Tajana” },
  “DJHa7I”: { id: “DJHa7I”, name: “Delfina” },
  “hEWmRa”: { id: “hEWmRa”, name: “Filip” },
  “Ck9YNg”: { id: “Ck9YNg”, name: “Amie },
  “AG7piI”: { id: “AG7piI”, name: “Radoslav” },
  “rtwj0V”: { id: “rtwj0V”, name: “Embla” },
  “VbnpoA”: { id: “VbnpoA”, name: “Natalia” }
}
```

```
students.find(student ⇒ {
  return student.id === 4
})
```

INDEXING

```
const students = [
  { id: “cjvbZq”, name: “Tajana” },
  { id: “DJHa7I”, name: “Delfina” },
  { id: “hEWmRa”, name: “Filip” },
  { id: “Ck9YNg”, name: “Amie” },
  { id: “AG7piI”, name: “Radoslav” },
  { id: “rtwj0V”, name: “Embla” },
  { id: “VbnpoA”, name: “Natalia” }
]
```

```
const students = {
  “cjvbZq”: { id: “cjvbZq”, name: “Tajana” },
  “DJHa7I”: { id: “DJHa7I”, name: “Delfina” },
  “hEWmRa”: { id: “hEWmRa”, name: “Filip” },
  “Ck9YNg”: { id: “Ck9YNg”, name: “Amie },
  “AG7piI”: { id: “AG7piI”, name: “Radoslav” },
  “rtwj0V”: { id: “rtwj0V”, name: “Embla” },
  “VbnpoA”: { id: “VbnpoA”, name: “Natalia” }
}
```

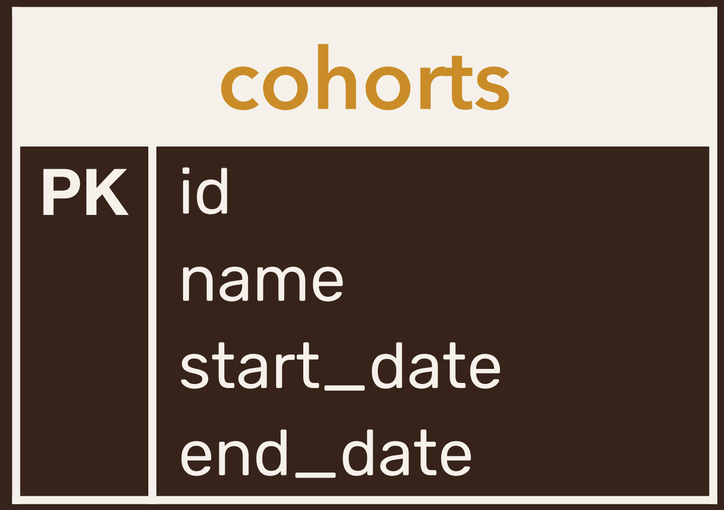
```
students.find(student ⇒ {
  return student.id === 4
})
```

```
students[‘4’]
```

QUERIES

>

```sql
CREATE TABLE cohorts (
  id SERIAL PRIMARY KEY NOT NULL,
  name VARCHAR(255) NOT NULL,
  start_date DATE,
  end_date DATE
);
```




```
cohort_id
INTEGER
REFERENCES cohorts(id)
ON DELETE CASCADE
```

CREATE	▶	INSERT
READ	▶	SELECT
UPDATE	▶	UPDATE
DELETE	▶	DELETE

INSERT (ONE)

```
INSERT
INTO cohorts (
  id,
  name,
  start_date,
  end_date
) VALUES (
  1,
  ‘FEB12',
  ‘2018-02-12T08:00:00.000Z',
  ‘2018-04-20T07:00:00.000Z'
);
```

```
INSERT INTO cohorts (id, name)
VALUES (1, ‘FEB12’),
       (2, ‘MAR12’),
       (3, ‘APR09’),
       (4, ‘MAY07’);
```

INSERT (MANY)

SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY

```
SELECT * FROM students;
```

students			
id	name	start_date	cohort_id
1	Tajana Meyrick	2018-02-12	1
2	Delfina Hayes	2018-03-12	2
3	Filip Bell	2018-03-12	2
4	Amie Fabbro	2018-04-09	3
5	Radoslav Pavlov	2018-04-09	3
6	Embla Bösch	2018-04-09	3
7	Natalia Armati	2018-04-09	3
8	Juraj Jansons	2018-04-09	3
9	Corona Headley	2018-05-07	4
10	Kwame Bernhardsson	2018-05-07	4
11	Helene Daubney	2018-05-07	4
12	Lucette Bianco	2018-05-07	4

```
SELECT id, name, cohort_id FROM students;
```

students		
id	name	cohort_id
1	Tajana Meyrick	1
2	Delfina Hayes	2
3	Filip Bell	2
4	Amie Fabbro	3
5	Radoslav Pavlov	3
6	Embla Bösch	3
7	Natalia Armati	3
8	Juraj Jansons	3
9	Corona Headley	4
10	Kwame Bernhardsson	4
11	Helene Daubney	4
12	Lucette Bianco	4

```sql
SELECT count(id) FROM students;
```

students
count
12

```sql
SELECT count(id) AS student_count
FROM students
WHERE cohort_id = 3;
```

students
count
5

```sql
SELECT
   cohort_id,
   count(id) AS student_count
FROM students
GROUP BY cohort_id;
```

students	
cohort_id	student_count
1	1
2	2
3	5
4	4

```sql
SELECT
   cohort_id,
   count(id) AS student_count
FROM students
GROUP BY cohort_id
HAVING count(id) > 1;
```

students	
cohort_id	student_count
2	2
3	5
4	4

```sql
SELECT
   cohort_id,
   count(id) AS student_count
FROM students
GROUP BY cohort_id
HAVING count(id) ⩾ 2
ORDER BY count(id) DESC;
```

students	
cohort_id	student_count
3	5
4	4
2	2

```
SELECT
   cohort_id,
   count(id) AS student_count
FROM students
WHERE start_date < ‘2018-05-01’
GROUP BY cohort_id
HAVING count(id) ⩾ 2
ORDER BY count(id) DESC;
```

students	
cohort_id	student_count
3	5
2	2

UPDATE
DELETE

```
UPDATE users SET password = ‘123456’;
DELETE FROM users;
```

ALWAYS USE WHERE

EVEN IF YOU
WANT TO
AFFECT THE
WHOLE TABLE

INNER
LEFT OUTER
RIGHT OUTER
FULL OUTER
CROSS

JOIN

AMBIGUITY

```
students.id,
students.name,
cohorts.name
```

INNER

students		
id	name	cohort_id
1	Sam Billings	1
2	Susan Hudson	null
3	Malloy Jenkins	3

+

cohorts	
id	name
1	FEB12
2	MAR12
3	APR09

id	student_name	cohort_name
1	Sam Billings	FEB12
3	Malloy Jenkins	APR09

INNER

```
SELECT
   students.id,
   students.name AS student_name,
   cohorts.name AS cohort_name
FROM students
INNER JOIN cohorts
ON cohorts.id = students.cohort_id;
```

OUTER

students		
id	name	cohort_id
1	Sam Billings	1
2	Susan Hudson	null
3	Malloy Jenkins	3

+

cohorts	
id	name
1	FEB12
2	MAR12
3	APR09

id	student_name	cohort_name
1	Sam Billings	FEB12
2	Susan Hudson	null
3	Malloy Jenkins	APR09

OUTER

```
SELECT
   students.id,
   students.name AS student_name,
   cohorts.name AS cohort_name
FROM students
LEFT OUTER JOIN cohorts
ON cohorts.id = students.cohort_id;
```

QUESTIONS?