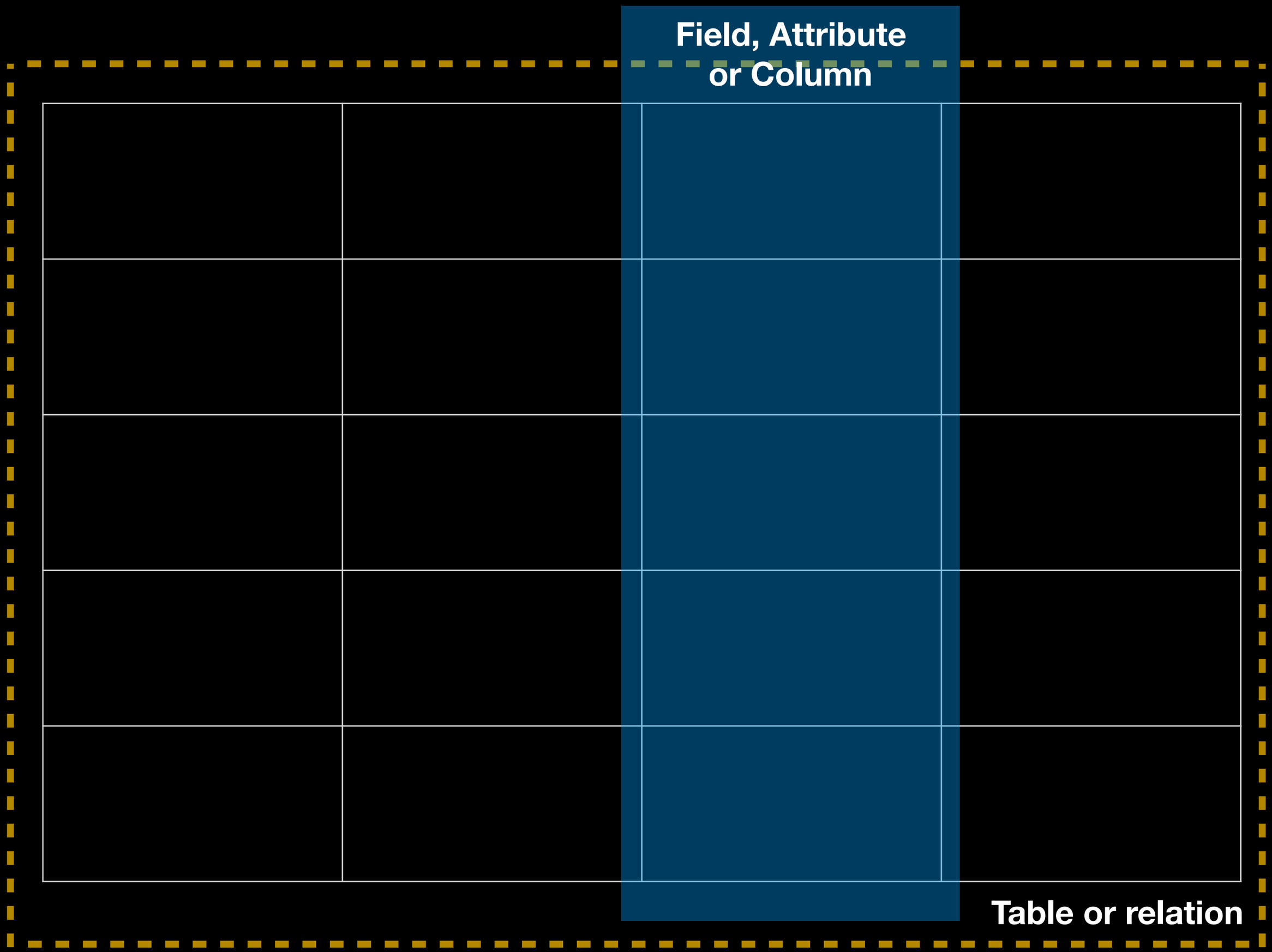
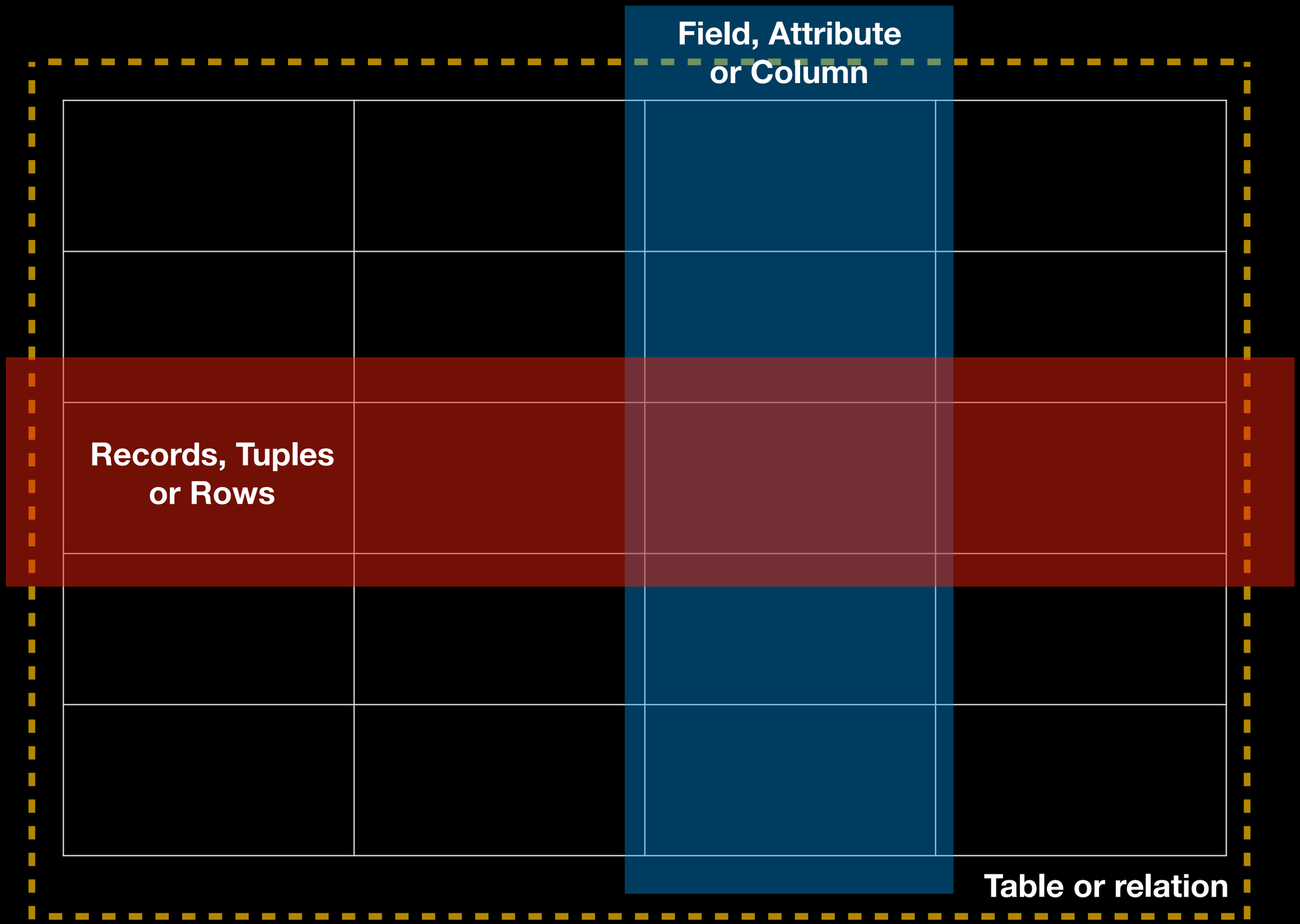
SQL

Structured Query Language


Table or relation


Field, Attribute
or Column
Table or relation


Field, Attribute
or Column
Records, Tuples
or Rows
Table or relation

Create
Read
Update
Delete

INSERT
SELECT
UPDATE
DELETE

CREATE TABLE table (column_name type, …);

DATA TYPES

BLOB
INTEGER
NUMERIC
REAL
TEXT

BLOB
INTEGER
    smallint
    integer
    bigint
NUMERIC
REAL
TEXT

BLOB
INTEGER
NUMERIC
REAL
    real
    double precision
TEXT

BLOB
INTEGER
NUMERIC
- boolean
- date
- datetime
- numeric(scale, precision)
- time
- timestamp

REAL
TEXT

BLOB
INTEGER
NUMERIC
REAL
TEXT
- char(n)
- varchar(n)
- text

Name	Storage Size	Description	Range
`smallint`	2 bytes	small-range integer	-32768 to +32767
`integer`	4 bytes	typical choice for integer	-2147483648 to +2147483647
`bigint`	8 bytes	large-range integer	-9223372036854775808 to +9223372036854775807
`decimal`	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
`numeric`	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
`real`	4 bytes	variable-precision, inexact	6 decimal digits precision
`double precision`	8 bytes	variable-precision, inexact	15 decimal digits precision
`smallserial`	2 bytes	small autoincrementing integer	1 to 32767
`serial`	4 bytes	autoincrementing integer	1 to 2147483647
`bigserial`	8 bytes	large autoincrementing integer	1 to 9223372036854775807

https://www.postgresql.org/docs/12/datatype-numeric.html

DDL VS DML

DATA

DEFINITION MANIPULATION

LANGUAGE

CREATE TABLE table (column_name type, …);

INSERT INTO table (column, …) VALUES (value, …);

DELETE FROM table WHERE condition;

https://www.postgresql.org/docs/7.4/ddl.html

```
CREATE TABLE buildings (
building_id INTEGER,
building_name VARCHAR(20),
height INTEGER );
```

https://www.postgresql.org/docs/7.4/dml.html

```
INSERT INTO buildings VALUES
(1, ‘World Trade Center’, 541);
```

https://www.postgresql.org/docs/7.4/dml.html

```
INSERT INTO buildings VALUES
(1, ‘World Trade Center’, 541);
```

https://www.postgresql.org/docs/7.4/dml.html

```
UPDATE buildings SET height =
height * 1.1 WHERE height < 100
```

https://www.postgresql.org/docs/7.4/dml.html

```
DELETE FROM buildings
WHERE height < 100
```

Queries

Queries

SELECT
WHERE
LIKE
LIMIT
GROUP BY
ORDER BY
JOIN
...

Queries

SELECT columns FROM table;

Queries

```
SELECT * FROM table;
```

Queries

SELECT column1 as a, columns2 as b FROM table;

Queries

```
SELECT * FROM buildings WHERE height > 100;
```

Queries

```
SELECT * FROM buildings LIMIT 5;
```

Queries

```
SELECT * FROM buildings ORDER BY height;
```

Queries

```
SELECT * FROM buildings ORDER BY height DESC;
```

Queries

AVG
COUNT
DISTINCT
MAX
MIN
...

Queries

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

Queries

```
SELECT MAX(height) FROM buildings;
```

Relational Databases
(RDBMS)

Date	Customer Name	Customer Phone Number	Order Total
2017-02-17	Bobby Tables	997-1009	$93.37
2017-02-18	Elaine Roberts	101-9973	$77.57
2017-02-20	Bobby Tables	997-1009	$99.73
2017-02-22	Bobby Tables	991-1009	$12.01

Relational Databases
(RDBMS)

orders

ID	Date	Customer	Amount
1	2017-02-17	37	$93.37
2	2017-02-18	73	$77.57
3	2017-02-20	37	$99.73
4	2017-02-22	37	$12.01

customers

ID	Name	Phone
37	Bobby Tables	997-1009
73	Elaine Roberts	101-9973

Relational Databases
(RDBMS)

orders

ID	Date	Customer	Amount
1	2017-02-17	37	$93.37
2	2017-02-18	73	$77.57
3	2017-02-20	37	$99.73
4	2017-02-22	37	$12.01

customers

ID	Name	Phone
37	Bobby Tables	997-1009
73	Elaine Roberts	101-9973

Primary key

Foreign key

computer scientists

ID	First	Last	Date of Birth	Nationality
21	Shafrira	Goldwasser	NULL	US
22	Alan	Turing	1912-05-23	UK
23	Judea	Pearl	1936-09-04	IL
24	Leslie	Lamport	1941-02-07	US
25	Michael	Stonebraker	1943–10-11	US
26	Whitfield	Diffie	1944-05-05	US
27	Martin	Hellman	1945-10-02	US
28	Silvio	Micali	1954–10-13	IT

winners

ID	Recipient	Year
58	23	2011
59	21	2012
60	28	2012
61	24	2013
62	25	2014
63	26	2015
64	27	2015

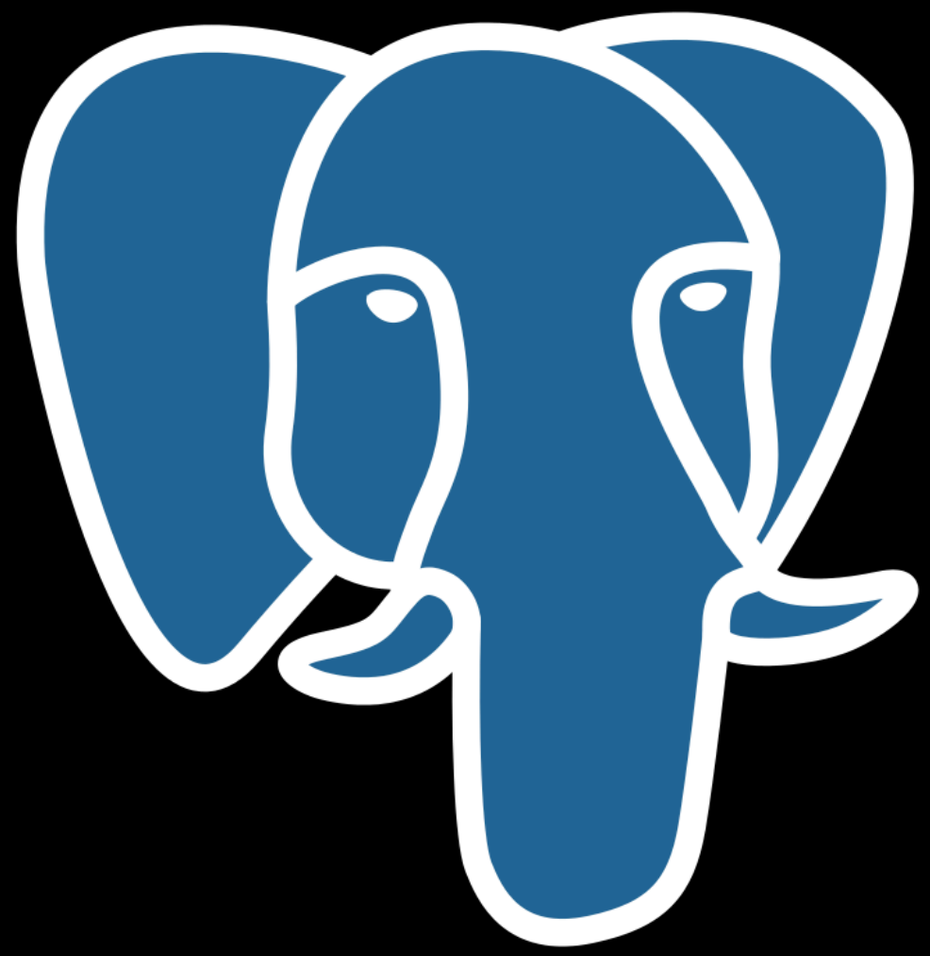
countries

ID	Country Name
IL	Israel
IT	Italy
UK	United Kingdom
US	United States

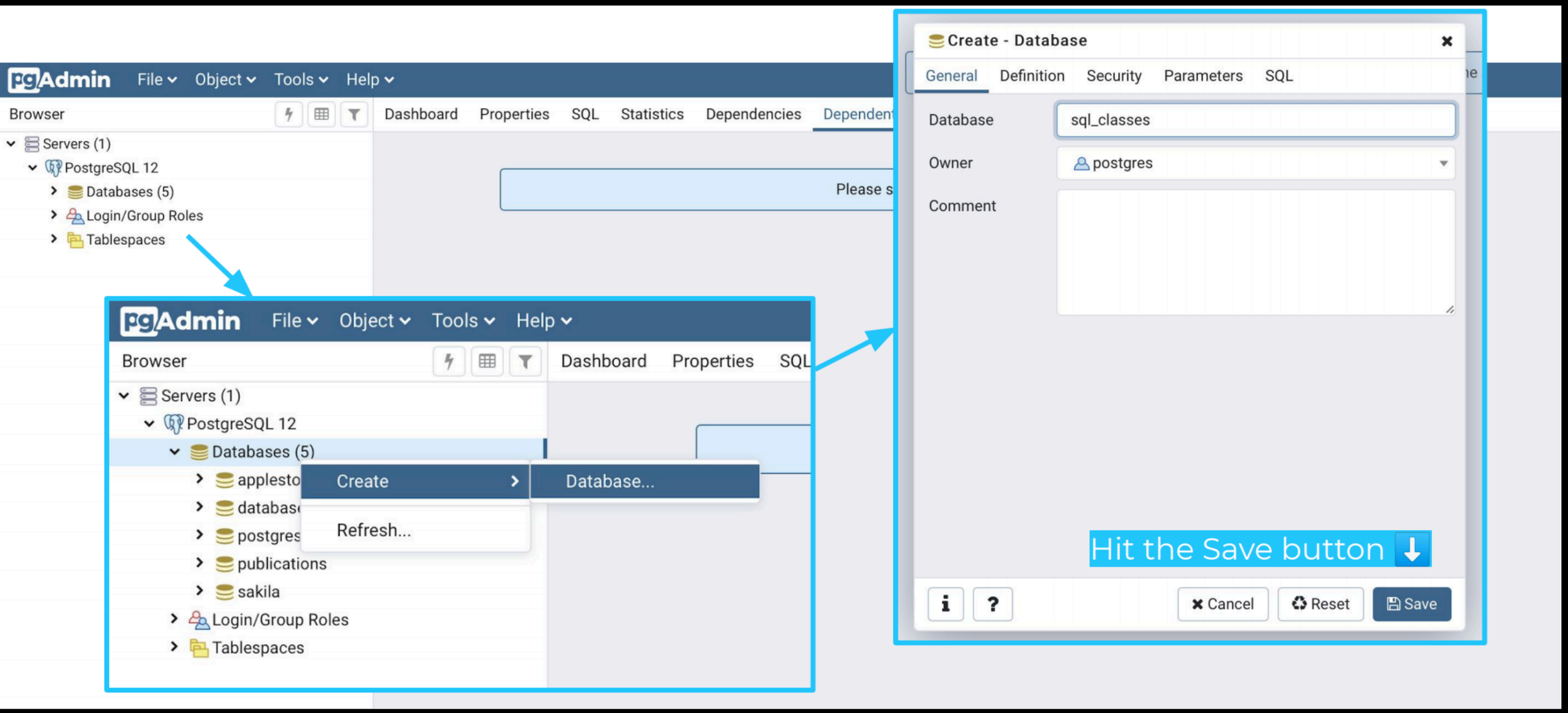
awards

ID	Main contribution
2011	Bayesian inference algorithms.
2012	Secure criptographic proofs.
2013	Distributed computing systems design.
2014	Database systems design.
2015	Diffie-Helmann key sharing.

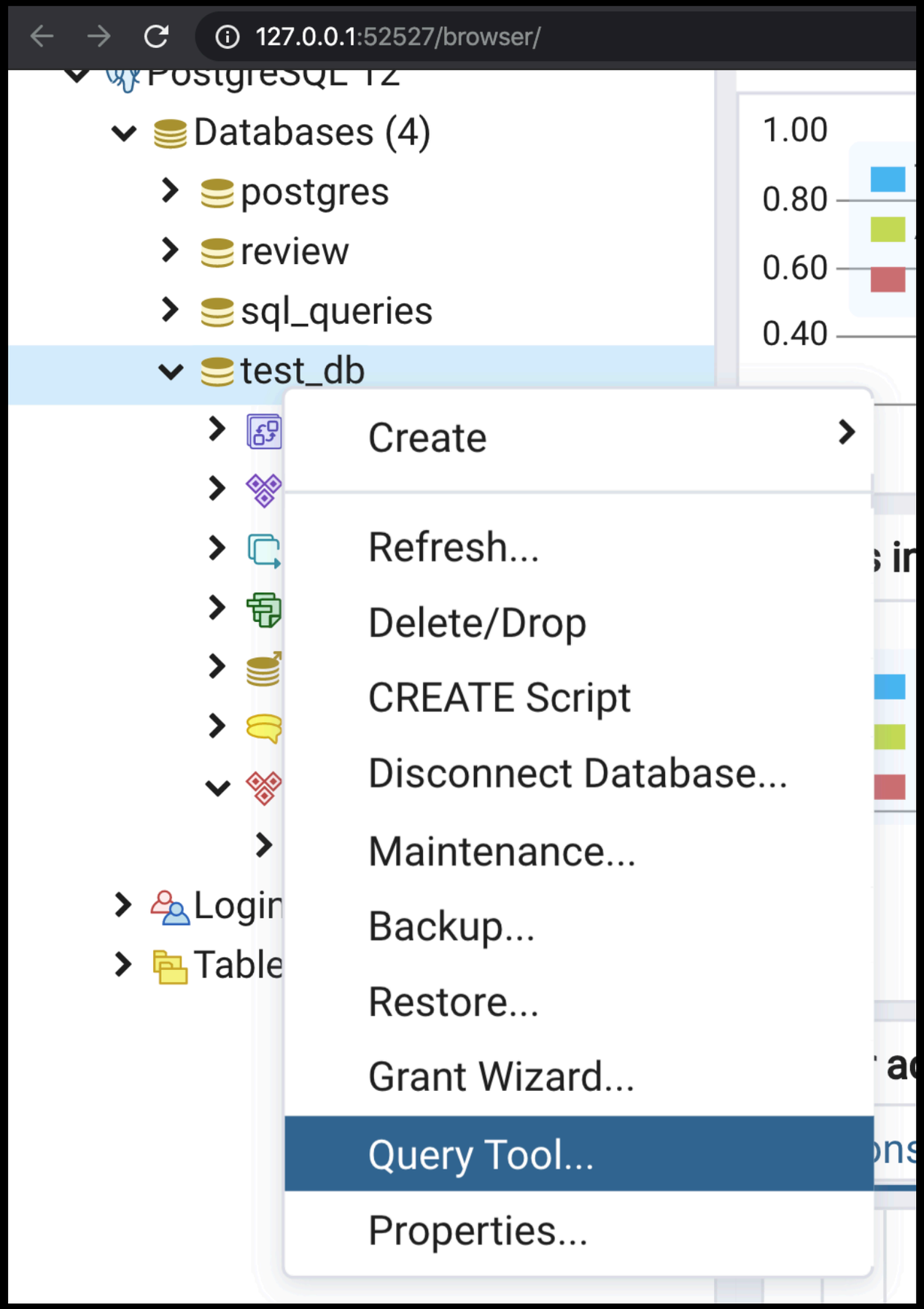
PostgreSQL



Creating a Database



Query Tool


127.0.0.1:52527/browser/
PostgreSQL 12
Databases (4)
postgres
review
sql_queries
test_db
Login
Table
1.00
0.80
0.60
0.40
Create
Refresh...
Delete/Drop
CREATE Script
Disconnect Database...
Maintenance...
Backup...
Restore...
Grant Wizard...
Query Tool...
Properties...

```sql
SELECT * FROM employees;
```

role text	name text	years_employed bigint	building text
Engineer	Becky A.	4	Burj Khalifa
Engineer	Dan B.	2	Burj Khalifa
Engineer	Sharon F.	6	Burj Khalifa
Engineer	Dan M.	4	Burj Khalifa
Engineer	Malcom…	1	Burj Khalifa
Artist	Tylar S.	2	Empire State
Artist	Sherma…	8	Empire State
Artist	Jakob J.	6	Empire State
Artist	Lillia A.	7	Empire State
Artist	Brandon…	7	Empire State
Manager	Scott K.	9	Burj Khalifa
Manager	Shirlee M.	3	Burj Khalifa
Manager	Daria O.	6	Empire State
Engineer	Yancy I.	0	[null]
Artist	Oliver P.	0	[null]

```
SELECT * FROM employees;
```

role text	name text	years_employed bigint	building text
Engineer	Becky A.	4	Burj Khalifa
Engineer	Dan B.	2	Burj Khalifa
Engineer	Sharon F.	6	Burj Khalifa
Engineer	Dan M.	4	Burj Khalifa
Engineer	Malcom...	1	Burj Khalifa
Artist	Tylar S.	2	Empire State
Artist	Sherma...	8	Empire State
Artist	Jakob J.	6	Empire State
Artist	Lillia A.	7	Empire State
Artist	Brandon...	7	Empire State
Manager	Scott K.	9	Burj Khalifa
Manager	Shirlee M.	3	Burj Khalifa
Manager	Daria O.	6	Empire State
Engineer	Yancy I.	0	[null]
Artist	Oliver P.	0	[null]

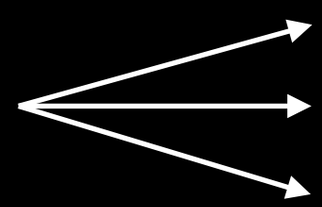
Relationships

One-to-one

When a record in a table is related to only one record in other table

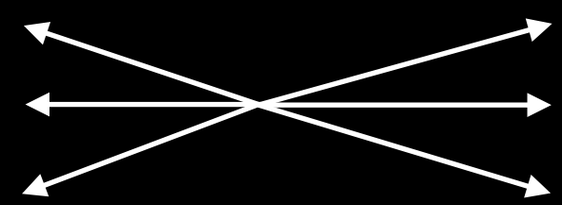
One-to-many

When a record in a table is related to two or more records in other table, but the opposite isn't true.

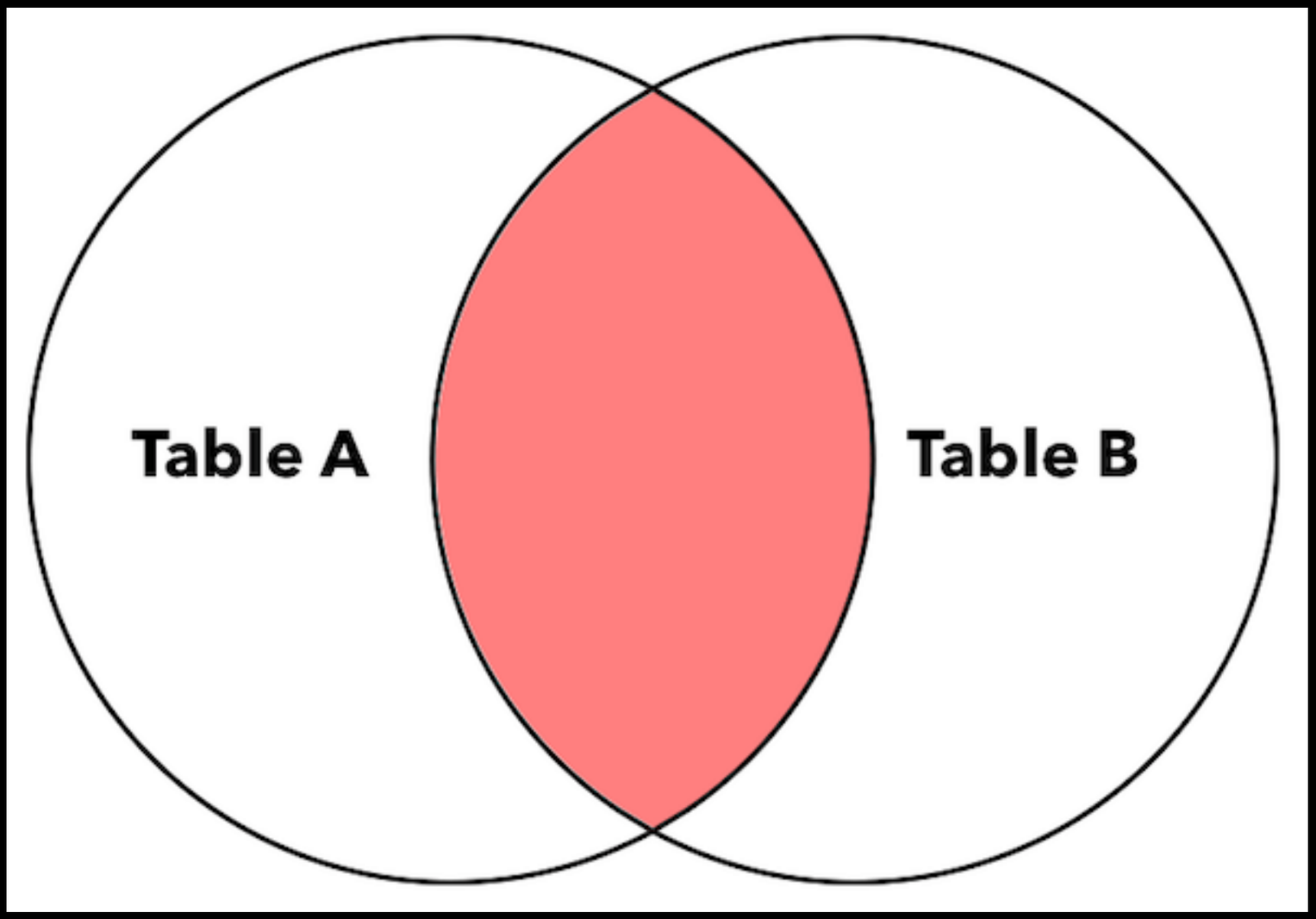


Many-to-many

When the records of a table are related to two or more records in other table, and the records of that table are related to two or more records in the former one.

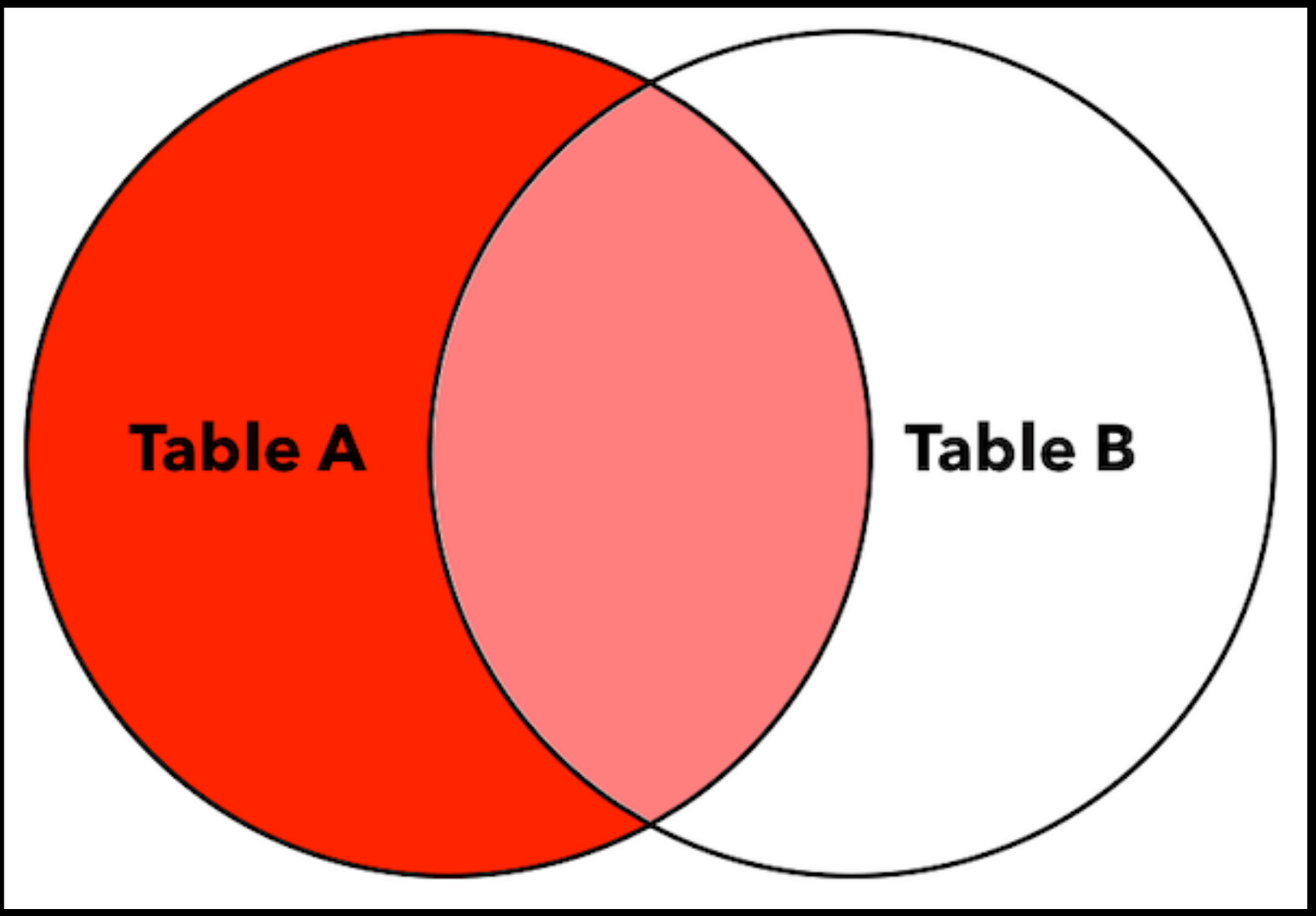


JOINS: INNER-JOIN



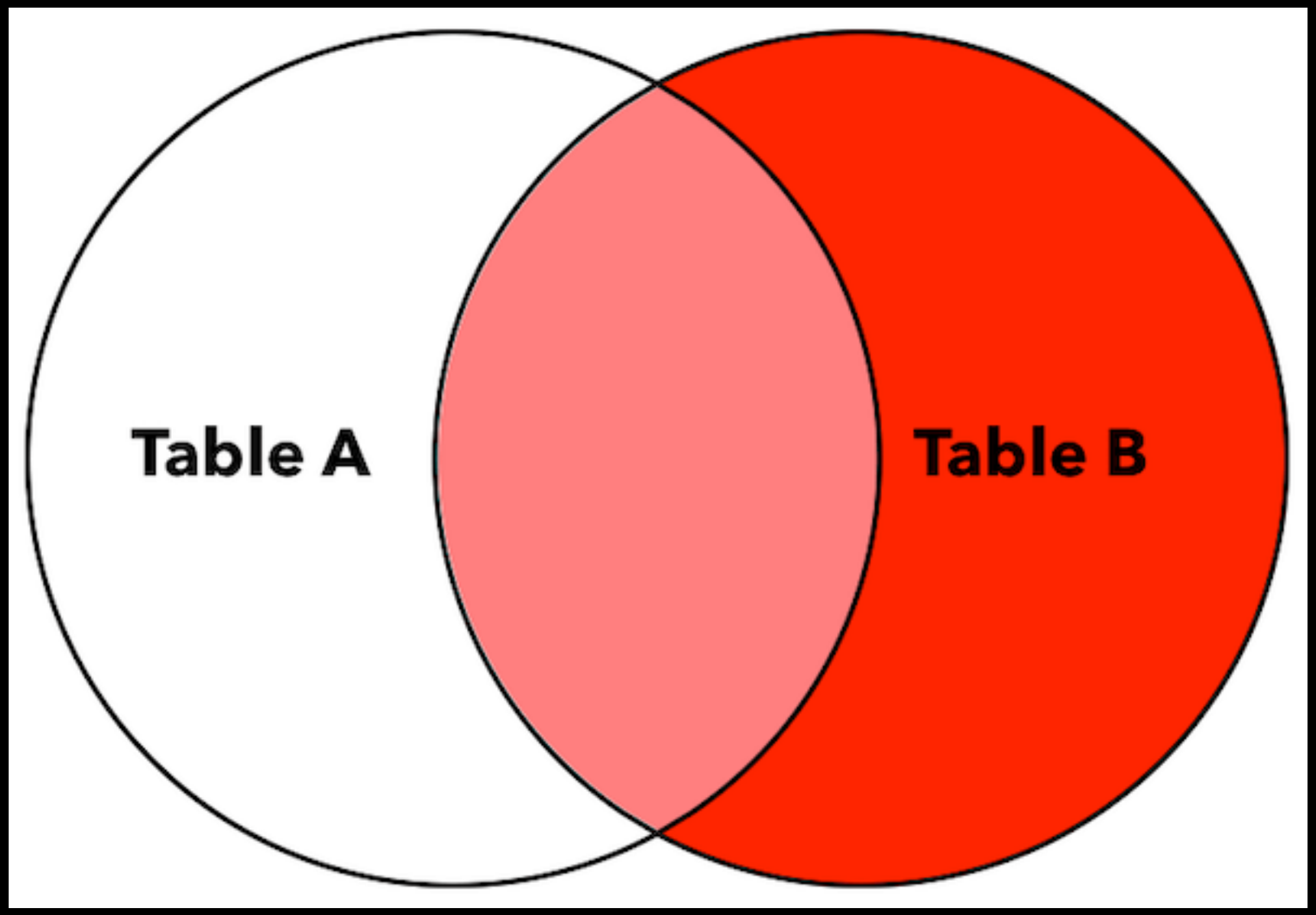

```
SELECT *
  FROM table_a
  INNER JOIN
  table_b
ON table_a.key = table_b.key
```

JOINS: LEFT-JOIN



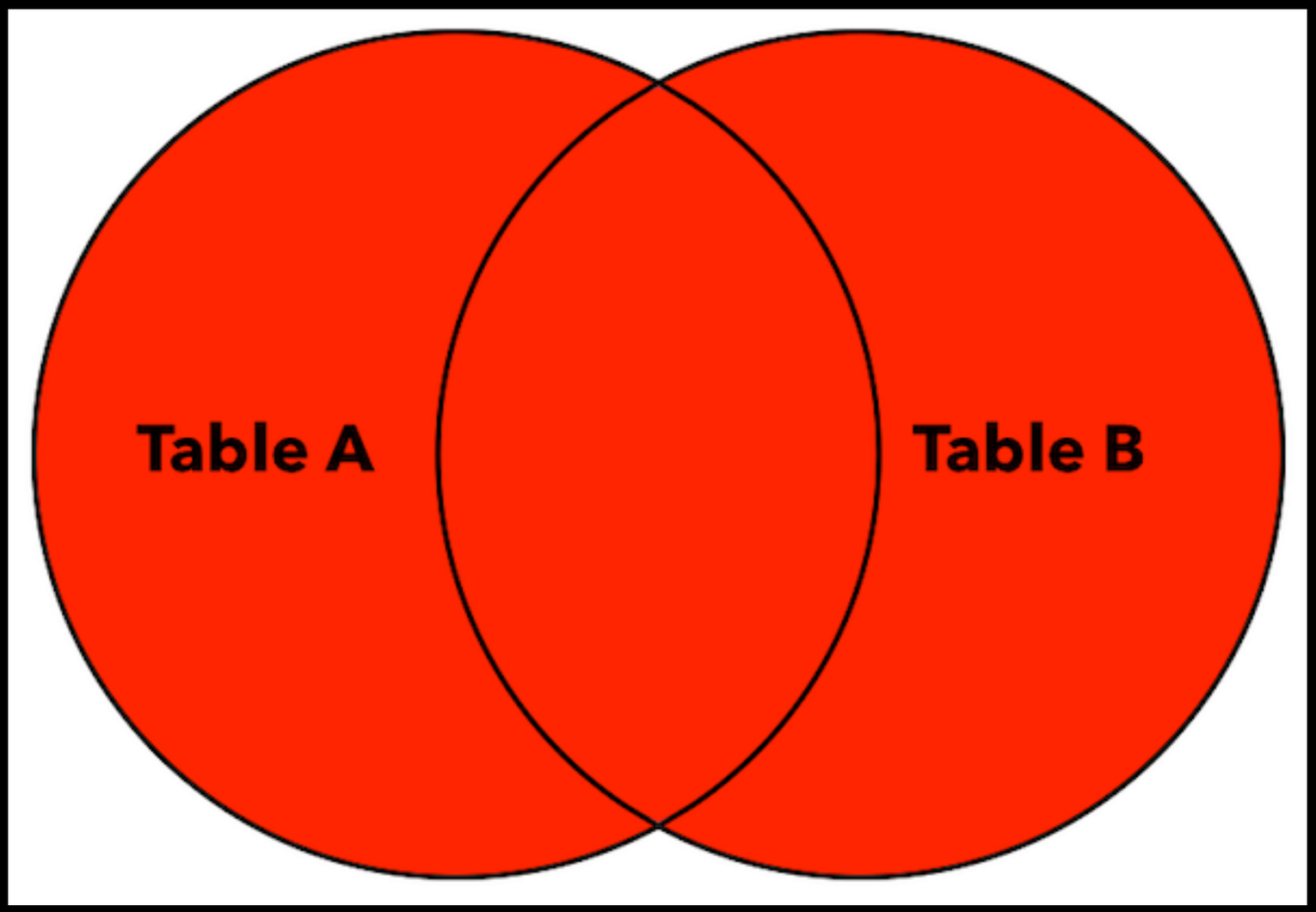

```
SELECT *
   FROM table_a
   LEFT JOIN
   table_b
ON table_a.key = table_b.key
```

JOINS: RIGHT-JOIN




```
SELECT *
   FROM table_a
   RIGHT JOIN
   table_b
ON table_a.key = table_b.key
```

JOINS: OUTER-JOIN

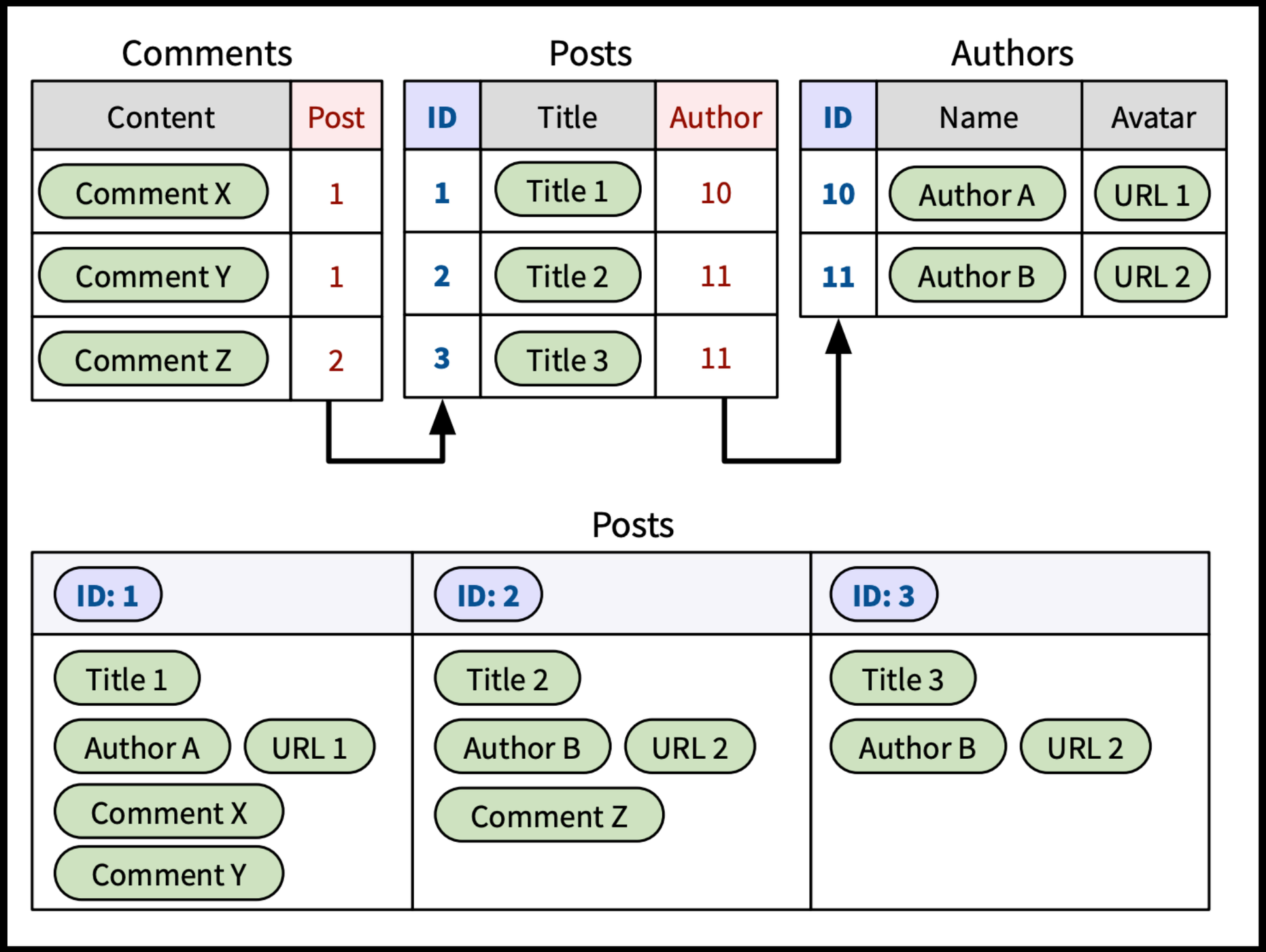



```
SELECT *
  FROM table_a
  FULL OUTER JOIN
  table_b
ON table_a.key = table_b.key
```

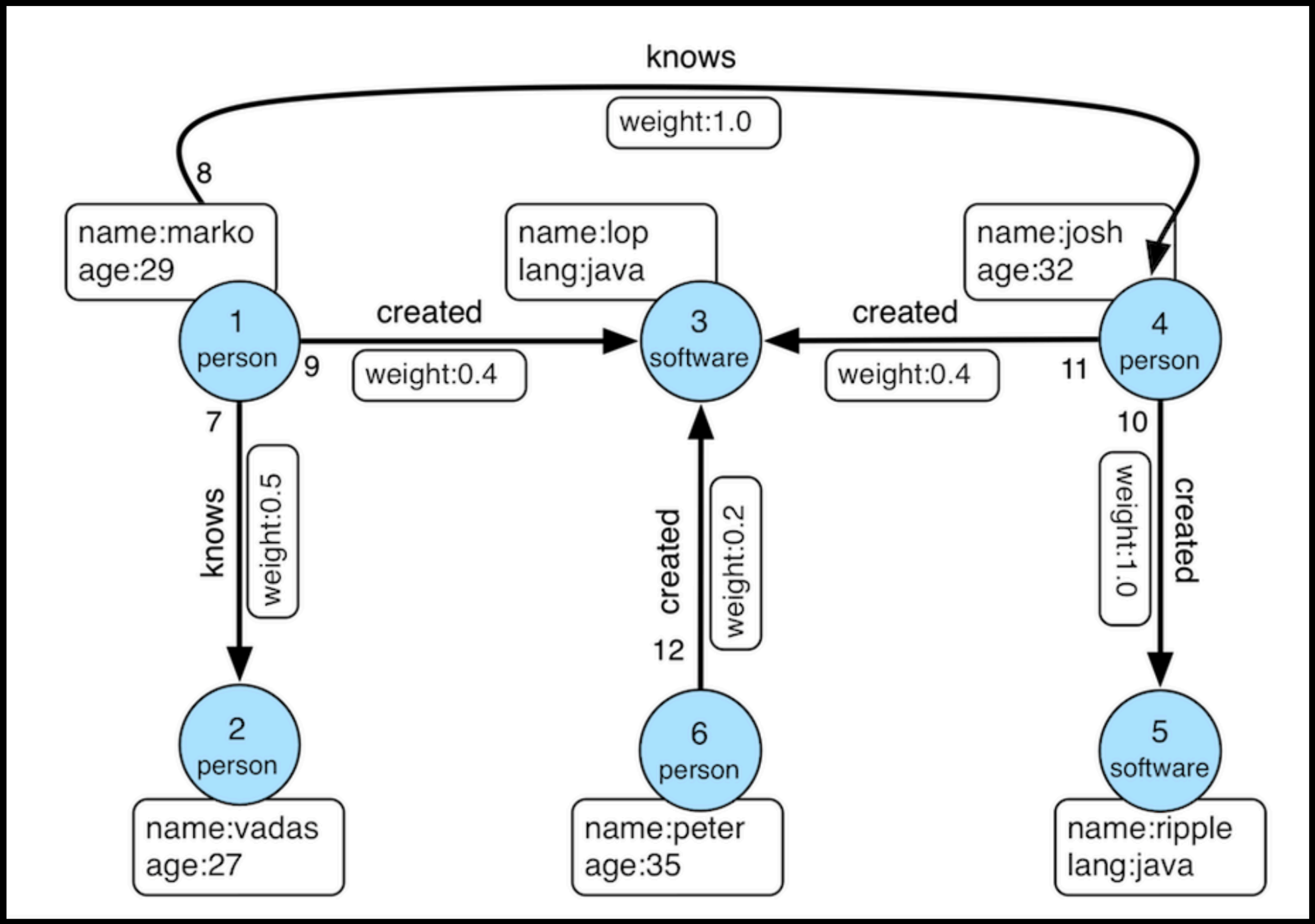
NoSQL

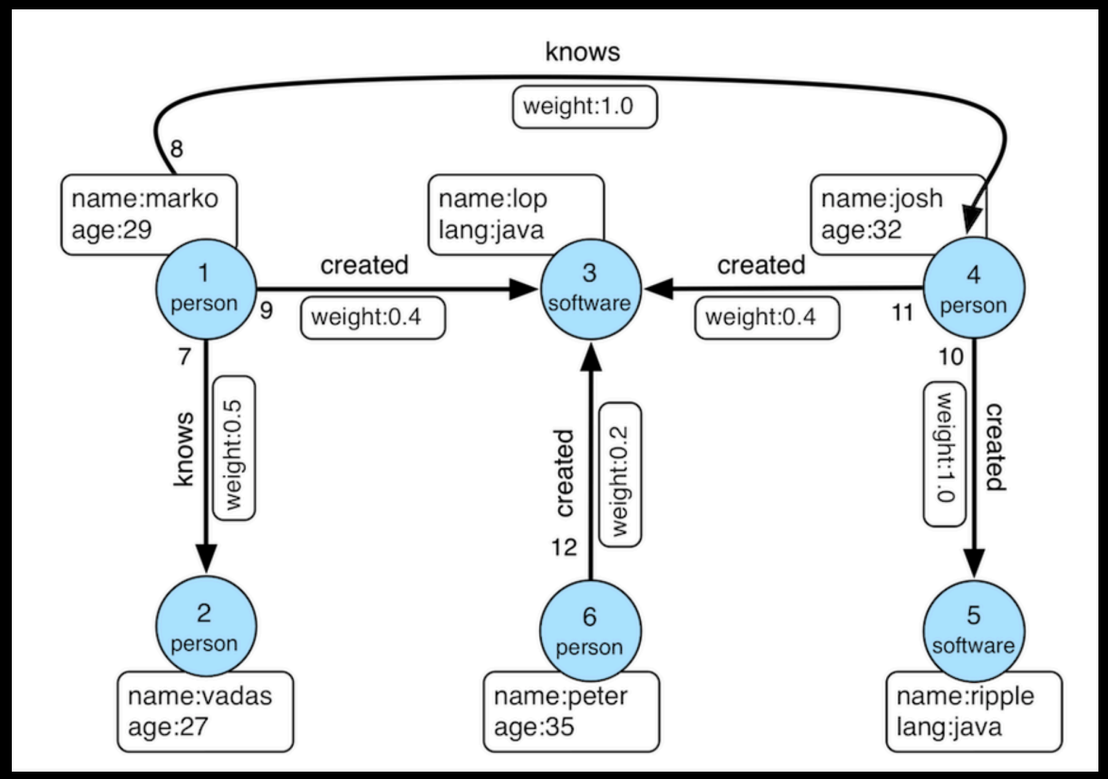

HOW TO WRITE A CV
DO YOU HAVE ANY EXPERTISE IN SQL?
NO
geek & poke
DOESN'T MATTER. WRITE: "EXPERT IN NO SQL"
Leverage the NoSQL boom

NoSQL: documents



Graph Databases





Marko's collaborators

```
g.V().has('person','name','marko').out('created')
```

```
g.V().has('person','name','marko').
            out('created').in('created').
            values('name')
```

```
g.V().has('person','name','marko').as('exclude').
            out('created').in('created').
            where(neq('exclude')).
            values('name')
```

Distributed Systems

Memory Hierarchy