



SQL

Cyrus Wong

origin	destination	duration
New York	London	415
Shanghai	Paris	760
Istanbul	Tokyo	700
New York	Paris	435
Moscow	Paris	245

PostgreSQL

PostgreSQL Data Types

- INTEGER
- SERIAL
- DECIMAL
- VARCHAR
- TIMESTAMP
- BOOLEAN
- ENUM
- ...

PostgreSQL Data Types

- INTEGER
- SERIAL
- DECIMAL
- VARCHAR
- TIMESTAMP
- BOOLEAN
- ENUM
- ...



0 vs NULL

CRUD

Create
Retrieve
Update
Delete

Create
Retrieve
Update
Delete

Create a Table

```
CREATE TABLE <table> (
    <colname1> <coltype1> (<attributes>),
    <colname2> <coltype2> (<attributes>),
    <colname3> <coltype3> (<attributes>),
    ...
);
```

origin	destination	duration
New York	London	415
Shanghai	Paris	760
Istanbul	Tokyo	700
New York	Paris	435
Moscow	Paris	245

origin	destination	duration
New York	London	415
Shanghai	Paris	760
Istanbul	Tokyo	700
New York	Paris	435
Moscow	Paris	245

```
CREATE TABLE flights (
    id SERIAL PRIMARY KEY,
    origin VARCHAR NOT NULL,
    destination VARCHAR NOT NULL,
    duration INTEGER NOT NULL
);
```

```
CREATE TABLE flights (
    id SERIAL PRIMARY KEY,
    origin VARCHAR NOT NULL,
    destination VARCHAR NOT NULL,
    duration INTEGER NOT NULL
);
```

```
CREATE TABLE flights (
    id SERIAL PRIMARY KEY,
    origin VARCHAR NOT NULL,
    destination VARCHAR NOT NULL,
    duration INTEGER NOT NULL
);
```

```
CREATE TABLE flights (
    id SERIAL PRIMARY KEY,
    origin VARCHAR NOT NULL,
    destination VARCHAR NOT NULL,
    duration INTEGER NOT NULL
);
```

```
CREATE TABLE flights (
    id SERIAL PRIMARY KEY,
    origin VARCHAR NOT NULL,
    destination VARCHAR NOT NULL,
    duration INTEGER NOT NULL
);
```

```
CREATE TABLE flights (
    id SERIAL PRIMARY KEY,
    origin VARCHAR NOT NULL,
    destination VARCHAR NOT NULL,
    duration INTEGER NOT NULL
);
```

Attributes (aka Constraints)

- NOT NULL
- UNIQUE
- PRIMARY KEY
- DEFAULT
- CHECK
- ...

Insert New Data

```
INSERT INTO <table>
(<col1>, <col2>, ... )
VALUES
(<val1>, <val2>, ... );
```

```
INSERT INTO flights  
(origin, destination, duration)  
VALUES  
( 'Lima' , 'New York' , 455);
```

```
INSERT INTO flights  
(origin, destination, duration)  
VALUES  
( 'Lima' , 'New York' , 455);
```

```
INSERT INTO flights  
(origin, destination, duration)  
VALUES  
( 'Lima' , 'New York' , 455);
```

```
INSERT INTO flights
(origin, destination, duration)
VALUES
('Lima', 'New York', 455);
```

```
INSERT INTO flights  
(origin, destination, duration)  
VALUES  
( 'Lima' , 'New York' , 455);
```

```
INSERT INTO flights  
(origin, destination, duration)  
VALUES  
( 'Lima' , 'New York' , 455);
```

```
INSERT INTO flights  
(origin, destination, duration)  
VALUES  
( 'Lima' , 'New York' , 455);
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

Create
Retrieve
Update
Delete

Selecting Data

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT origin, destination FROM flights;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT origin, destination FROM flights;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE id = 3;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE id = 3;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE id = 3;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE origin = 'New York';
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE origin = 'New York';
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE duration > 500;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE duration > 500;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE destination = 'Paris' AND duration > 500;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE destination = 'Paris' AND duration > 500;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE destination = 'Paris' OR duration > 500;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE destination = 'Paris' OR duration > 500;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE origin IN ('New York', 'Lima');
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE origin IN ('New York', 'Lima');
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE origin IN ('New York', 'Lima');
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE origin LIKE '%a%';
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE origin LIKE '%a%';
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE origin LIKE '%a%';
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights  
WHERE origin LIKE '%a%';
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

Some Useful SQL Functions

- SUM
- COUNT
- MIN
- MAX
- AVG
- ...

Create
Retrieve
Update
Delete

Updating Data

```
UPDATE <table>
SET <col> = <val>
WHERE <condition>;
```

```
UPDATE flights  
SET duration = 440  
WHERE origin = 'New York'  
AND destination = 'London';
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

id	origin	destination	duration
1	New York	London	440
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

Create
Retrieve
Update
Delete

Deleting Data

```
DELETE FROM <table>  
WHERE <condition>;
```

```
DELETE FROM flights  
WHERE destination = 'Tokyo';
```

id	origin	destination	duration
1	New York	London	440
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

id	origin	destination	duration
1	New York	London	440
2	Shanghai	Paris	760
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

Helpful Selector Clauses

- LIMIT
- ORDER BY
- GROUP BY
- HAVING
- ...

Foreign Keys

flights

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

flights

id	origin	orig_code	destination	dest_code	duration
1	New York	JFK	London	LHR	415
2	Shanghai	PVG	Paris	CDG	760
3	Istanbul	IST	Tokyo	NRT	700
4	New York	JFK	Paris	CDG	435
5	Moscow	SVO	Paris	CDG	245
6	Lima	LIM	New York	JFK	455

locations

id	name	code
1	New York	JFK
2	Shanghai	PVG
3	Istanbul	IST
4	London	LHR
5	Moscow	SVO
6	Lima	LIM
7	Paris	CDG
8	Tokyo	NRT

flights

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

flights

id	orig_id	dest_id	duration
1	1	4	415
2	2	7	760
3	3	8	700
4	1	7	435
5	5	7	245
6	6	1	455

passengers

id	name	flight_id
1	Alice	1
2	Bob	1
3	Charlie	2
4	Diane	2
5	Eric	4
6	Francine	6
7	Greg	6

Joining Tables

Types of Joins

- JOIN (aka INNER JOIN)
- LEFT JOIN (aka LEFT OUTER JOIN)
- RIGHT JOIN (aka RIGHT OUTER JOIN)
- FULL OUTER JOIN

Right Outer Join

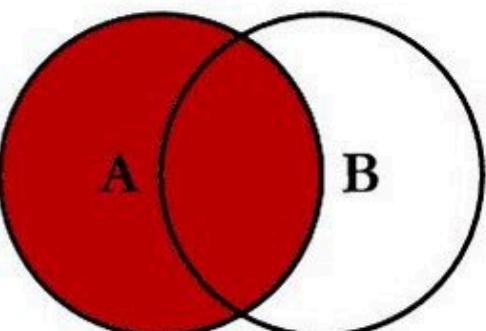


Left Outer Join

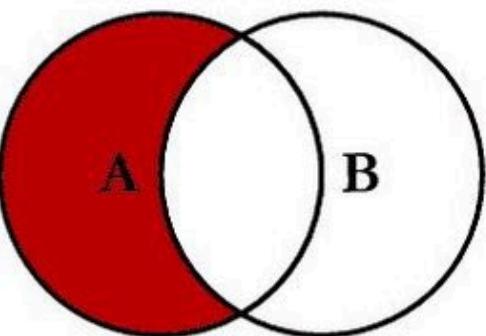
Full Outer Join



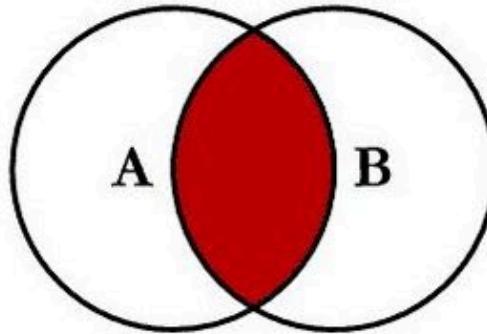
SQL JOINS



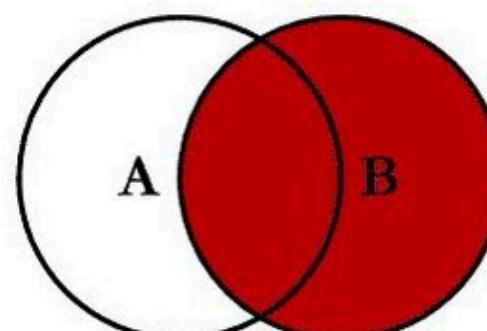
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



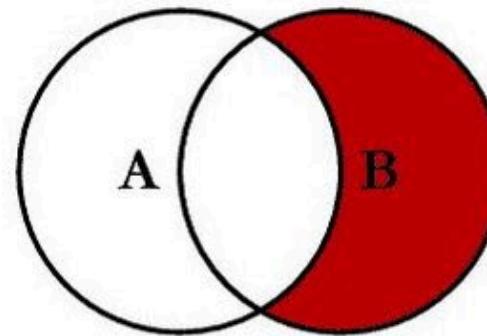
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



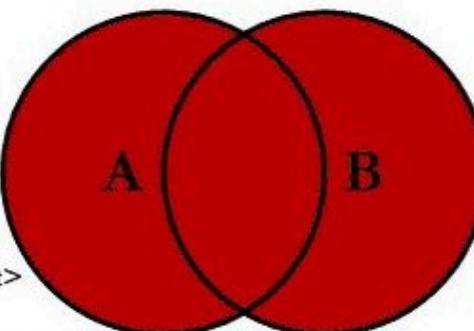
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



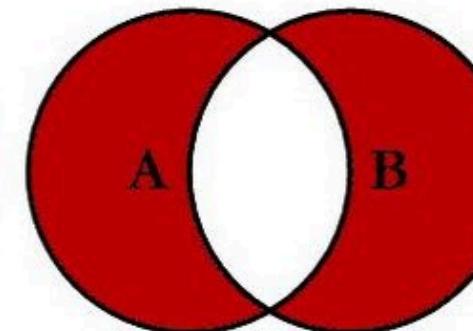
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

CREATE INDEX

Nested Queries

flights

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

passengers

id	name	flight_id
1	Alice	1
2	Bob	1
3	Charlie	2
4	Diane	2
5	Eric	4
6	Francine	6
7	Greg	6

```
SELECT flight_id FROM passengers  
GROUP BY flight_id HAVING COUNT(*) > 1;
```

```
SELECT flight_id FROM passengers  
GROUP BY flight_id HAVING COUNT(*) > 1;
```

id	name	flight_id
1	Alice	1
2	Bob	1
3	Charlie	2
4	Diane	2
5	Eric	4
6	Francine	6
7	Greg	6

```
SELECT flight_id FROM passengers  
GROUP BY flight_id HAVING COUNT(*) > 1;
```

id	name	flight_id
1	Alice	1
2	Bob	1
3	Charlie	2
4	Diane	2
5	Eric	4
6	Francine	6
7	Greg	6

flight_id
1
2
6

```
SELECT flight_id FROM passengers  
GROUP BY flight_id HAVING COUNT(*) > 1;
```

flight_id
1
2
6

```
SELECT * FROM flights WHERE id IN  
(SELECT flight_id FROM passengers  
GROUP BY flight_id HAVING COUNT(*) > 1);
```

flight_id
1
2
6

```
SELECT * FROM flights WHERE id IN  
(SELECT flight_id FROM passengers  
GROUP BY flight_id HAVING COUNT(*) > 1);
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

flight_id
1
2
6

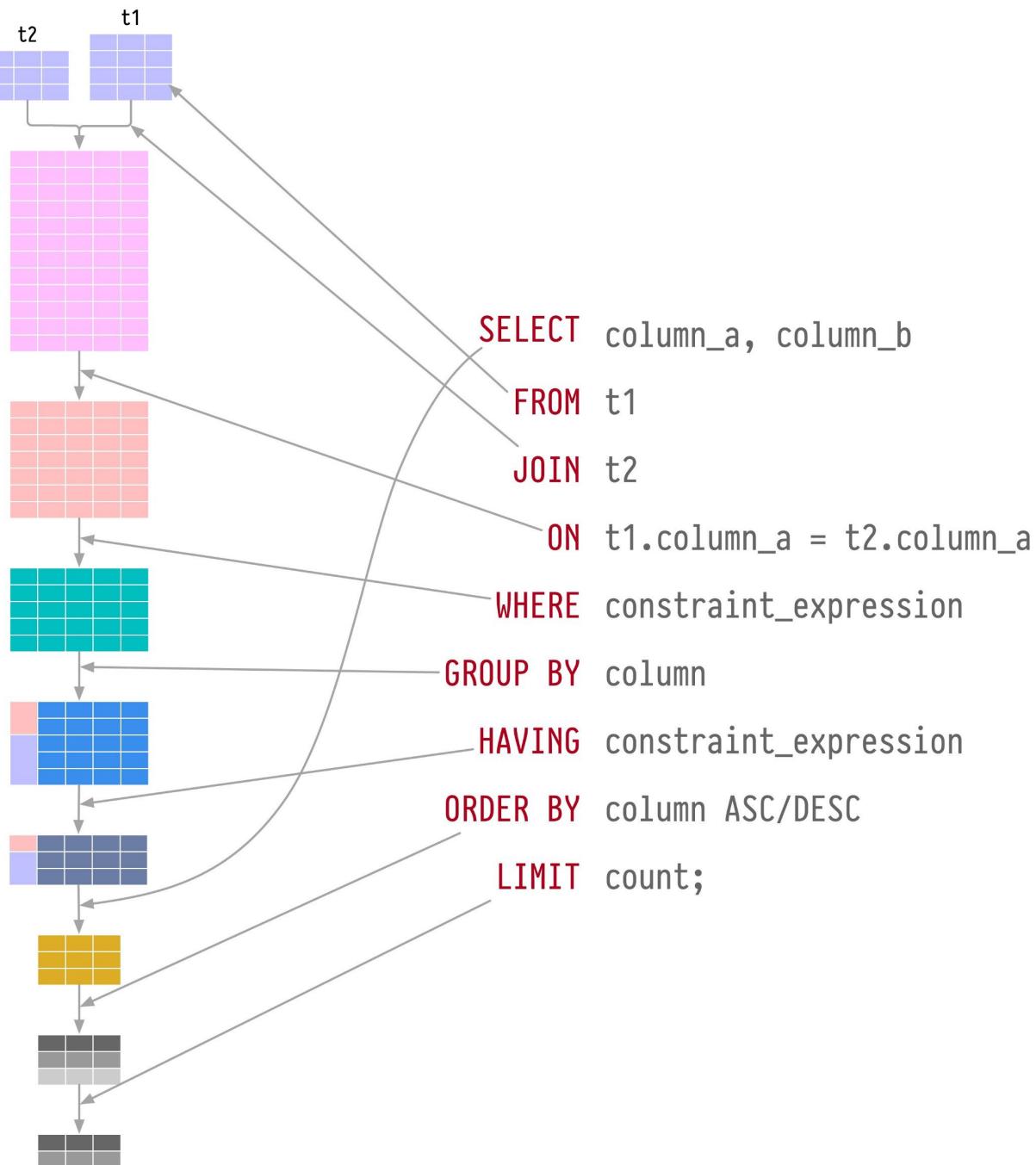
```

SELECT * FROM flights WHERE id IN
(SELECT flight_id FROM passengers
GROUP BY flight_id HAVING COUNT(*) > 1);

```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

flight_id
1
2
6



SQL Injection Attacks

username,

Enter Username

Password

Enter Password

Login

```
SELECT * FROM users  
WHERE (username = uname)  
AND (password = pword)
```

Username

milo

Password

.....

Login

```
SELECT * FROM users  
WHERE (username = uname)  
AND (password = pword)
```

```
SELECT * FROM users  
WHERE (username = 'milo')  
AND (password = 'puppy')
```

Username

hacker

Password

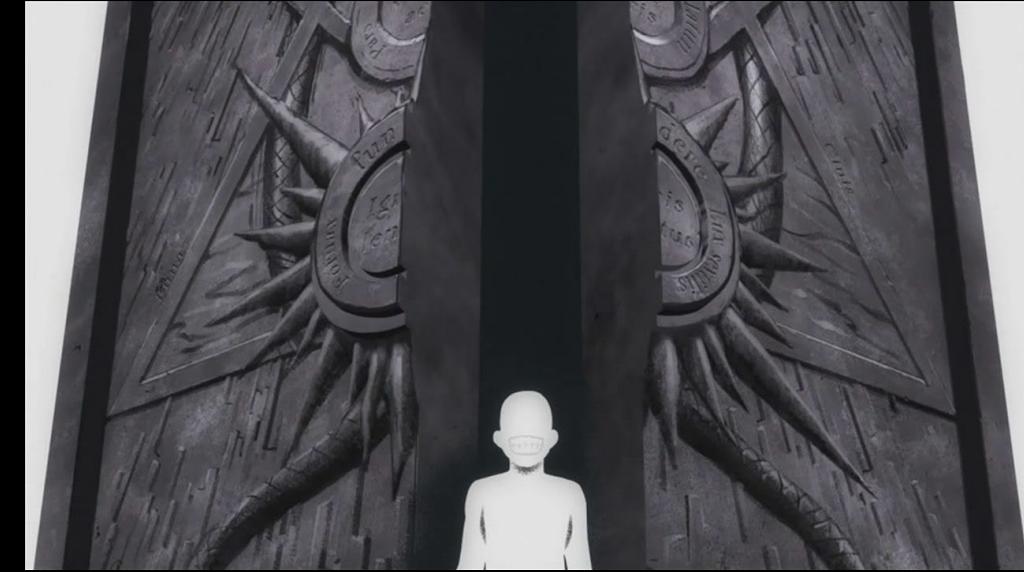
1' OR '1' = '1

Login

```
SELECT * FROM users  
WHERE (username = uname)  
AND (password = pword)
```

真理之門

```
SELECT * FROM users  
WHERE (username = 'hacker')  
AND (password = '1' OR '1' = '1')
```



SQL Injection.

User-Id : itswadeph

Password : newpassword

```
select * from Users where user_id= 'itswadeph'  
        and password = ' newpassword '
```

User-Id : ' OR 1= 1; /*

Password : */-

```
select * from Users where user_id= '' OR 1 = 1; /* '  
        and password = ' */-
```

HI, THIS IS
YOUR SON'S SCHOOL.
WE'RE HAVING SOME
COMPUTER TROUBLE.



OH, DEAR - DID HE
BREAK SOMETHING?
IN A WAY -)



DID YOU REALLY
NAME YOUR SON
Robert'); DROP
TABLE Students;-- ?



OH, YES. LITTLE
BOBBY TABLES,
WE CALL HIM.

WELL, WE'VE LOST THIS
YEAR'S STUDENT RECORDS.
I HOPE YOU'RE HAPPY.



AND I HOPE
YOU'VE LEARNED
TO SANITIZE YOUR
DATABASE INPUTS.

Her son name is
「Robert'); DROP TABLE students;--」

Result in all student's data loss!

IT狗攻略(SQL injection)



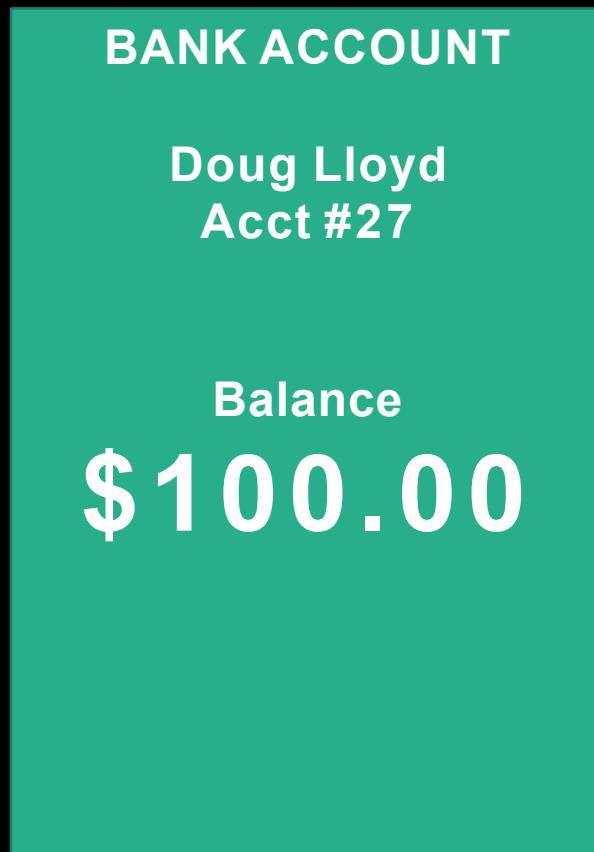
Don't try to write SQL by
appending String!



You should use ORM (Just write a
class!) to save the pain!

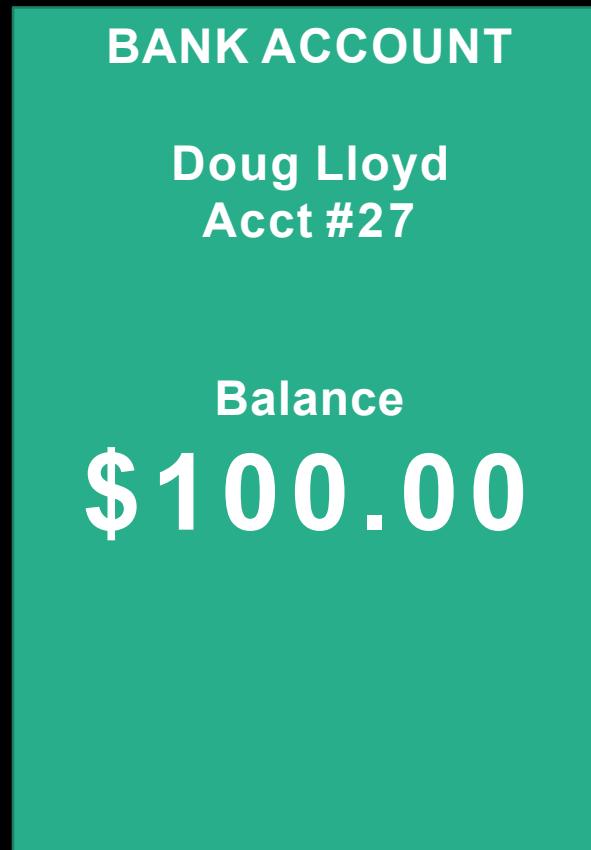
Race Conditions

Race Conditions



Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```



Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00



Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

```
UPDATE bank  
SET balance = balance - 100  
WHERE acct = 27
```



Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

```
UPDATE bank  
SET balance = balance - 100  
WHERE acct = 27
```



BANK ACCOUNT

Doug Lloyd
Acct #27

Balance

\$0.00

Race Conditions



Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```



Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

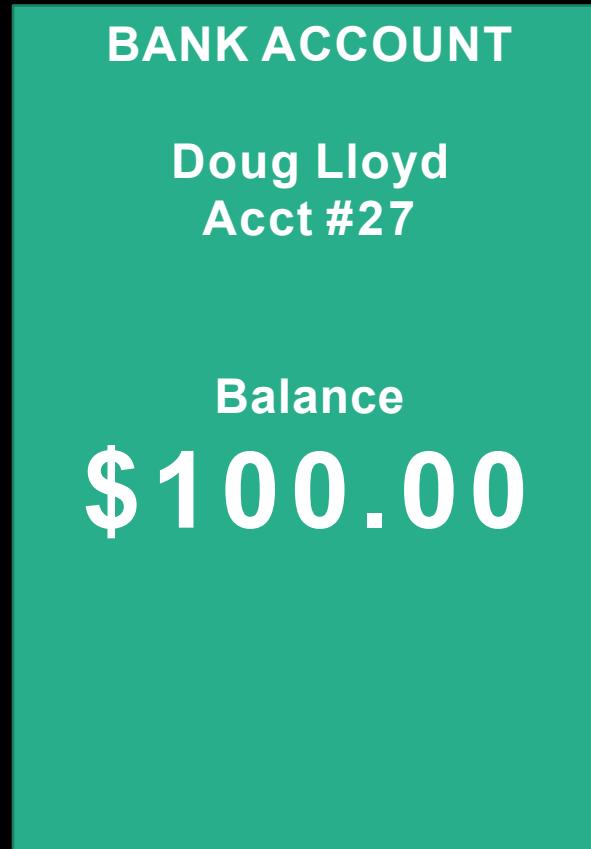
100.00



Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

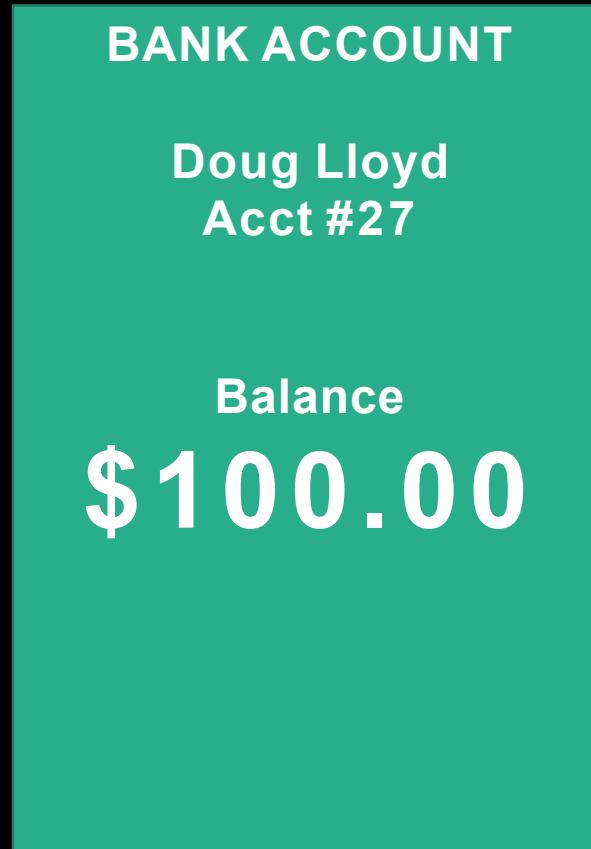


```
SELECT balance FROM bank WHERE  
acct = 27;
```

Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00



```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

```
UPDATE bank  
SET balance = balance - 100  
WHERE acct = 27
```



```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

```
UPDATE bank  
SET balance = balance - 100  
WHERE acct = 27
```



BANK ACCOUNT

Doug Lloyd
Acct #27

Balance

\$0.00

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

```
UPDATE bank  
SET balance = balance - 100  
WHERE acct = 27
```



BANK ACCOUNT

Doug Lloyd
Acct #27

Balance

\$0.00

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

```
UPDATE bank  
SET balance = balance - 100  
WHERE acct = 27
```

Race Conditions

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

```
UPDATE bank  
SET balance = balance - 100  
WHERE acct = 27
```



BANK ACCOUNT

Doug Lloyd
Acct #27

Balance

-\$100.00

```
SELECT balance FROM bank WHERE  
acct = 27;
```

100.00

```
UPDATE bank  
SET balance = balance - 100  
WHERE acct = 27
```



Transactions

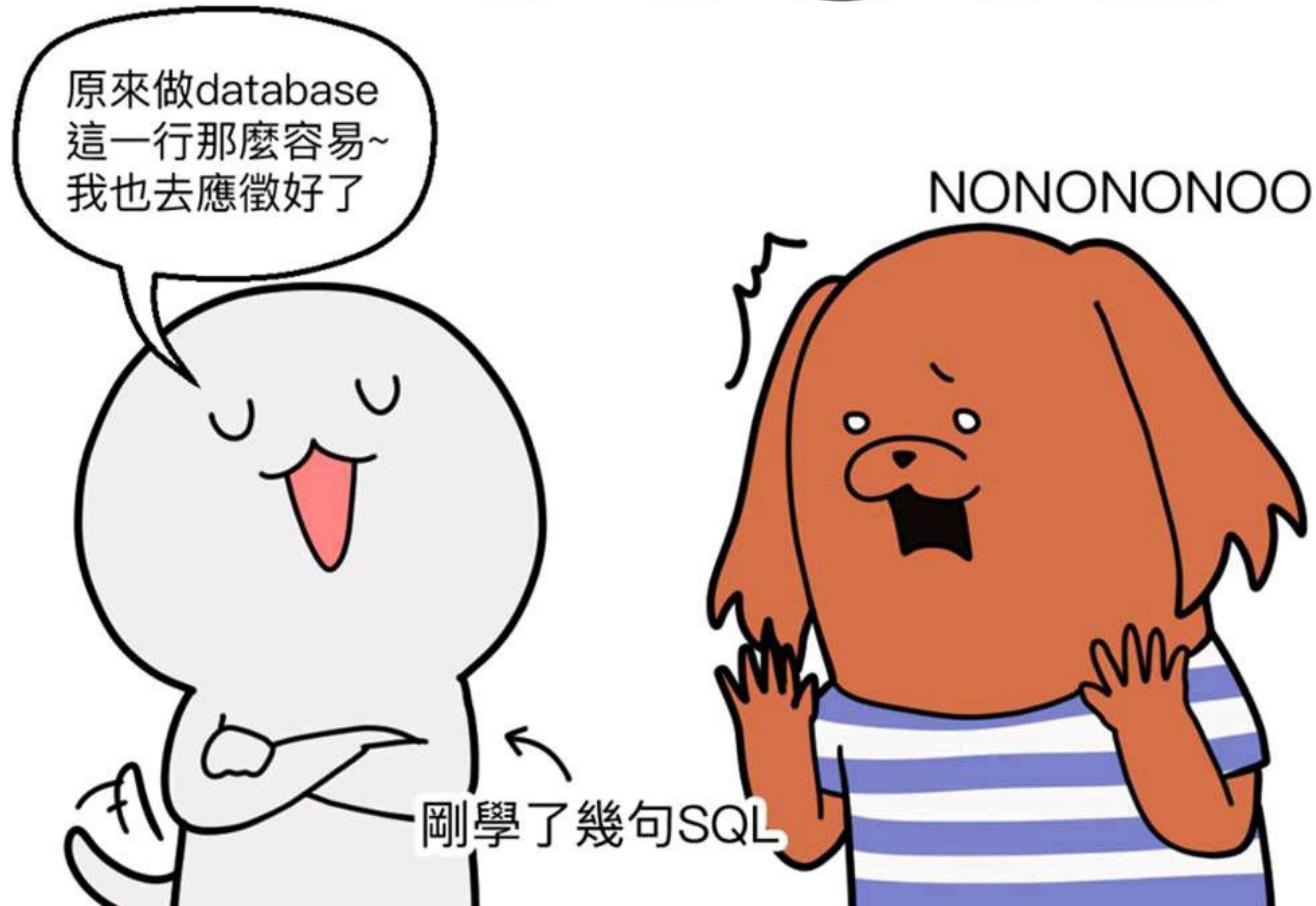
- BEGIN
- COMMIT

ACID

ACID (Atomicity, Consistency, Isolation, Durability) is a set of properties of database transactions intended to guarantee validity even in the event of errors, power failures, etc. In the context of databases, a sequence of database operations that satisfies the ACID properties (and these can be perceived as a single logical operation on the data) is called a transaction. For example, a transfer of funds from one bank account to another, even involving multiple changes such as debiting one account and crediting another, is a single transaction.

超淺談資料庫管理系統(DBMS)中
數據庫事務
正確執行的
基本要素 ↗

ACID



前設

柴柴的結餘:

\$500

唐唐的結餘:

\$500

條件:

結餘不能為負數



唐唐的結餘:

\$500

+500

Database 會順序處理臘臘 Transaction 內的指令。

Database 會順序處理臘臘 Transaction 內的指令。

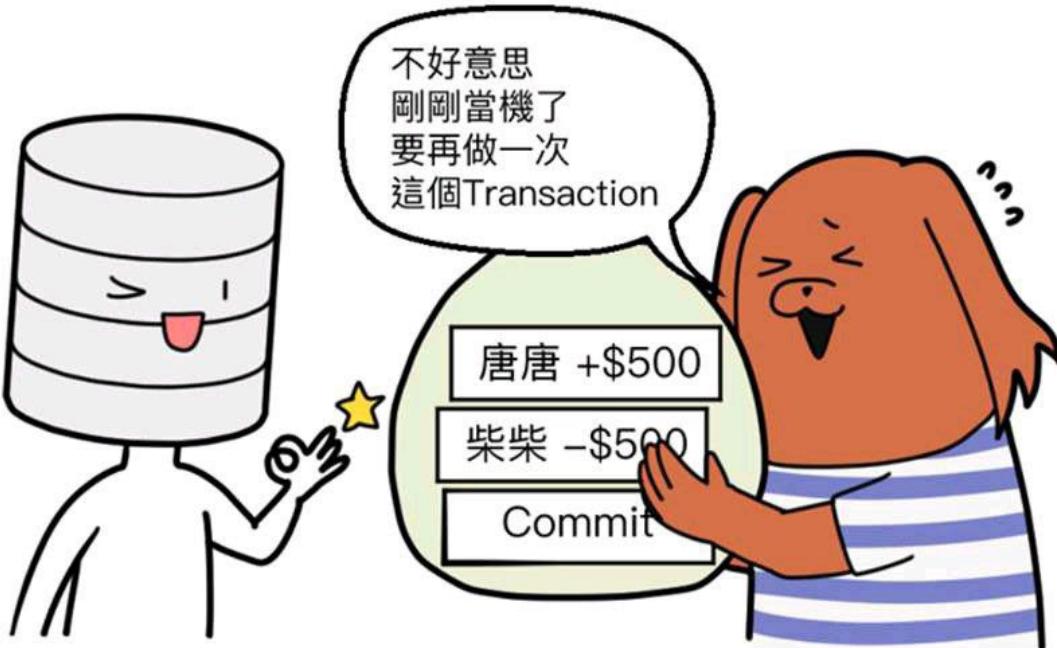
Database 會順序處理臘臘 Transaction 內的指令。

Atomocity就是保證資料不會 結束在中間的環節

柴柴的結餘:
\$500

唐唐的結餘:
\$500

因為還未完成
Transaction
所以結餘被
Rollback了



如果沒有做到Atomicity
唐唐的結餘就會出錯了

Transactions are often composed of multiple statements. Atomicity guarantees that each transaction is treated as a single "unit", which either succeeds completely, or fails completely: if any of the statements constituting a transaction fails to complete, the entire transaction fails and the database is left unchanged. An atomic system must guarantee atomicity in each and every situation, including power failures, errors and crashes.

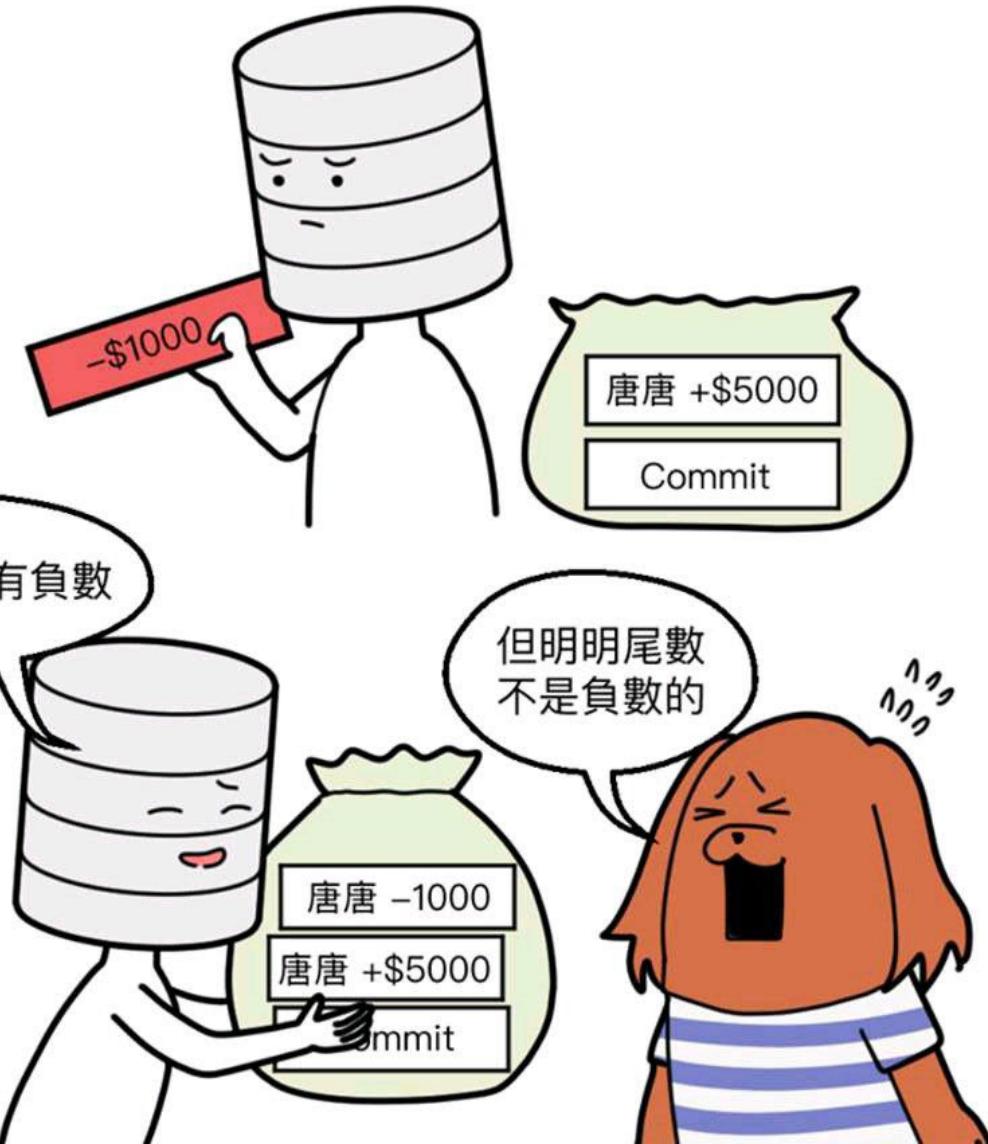
Consistency就是為了保障資料的 要附合了預設條件才可以Commit

唐唐的結餘:
\$500

條件:
結餘不能為負數

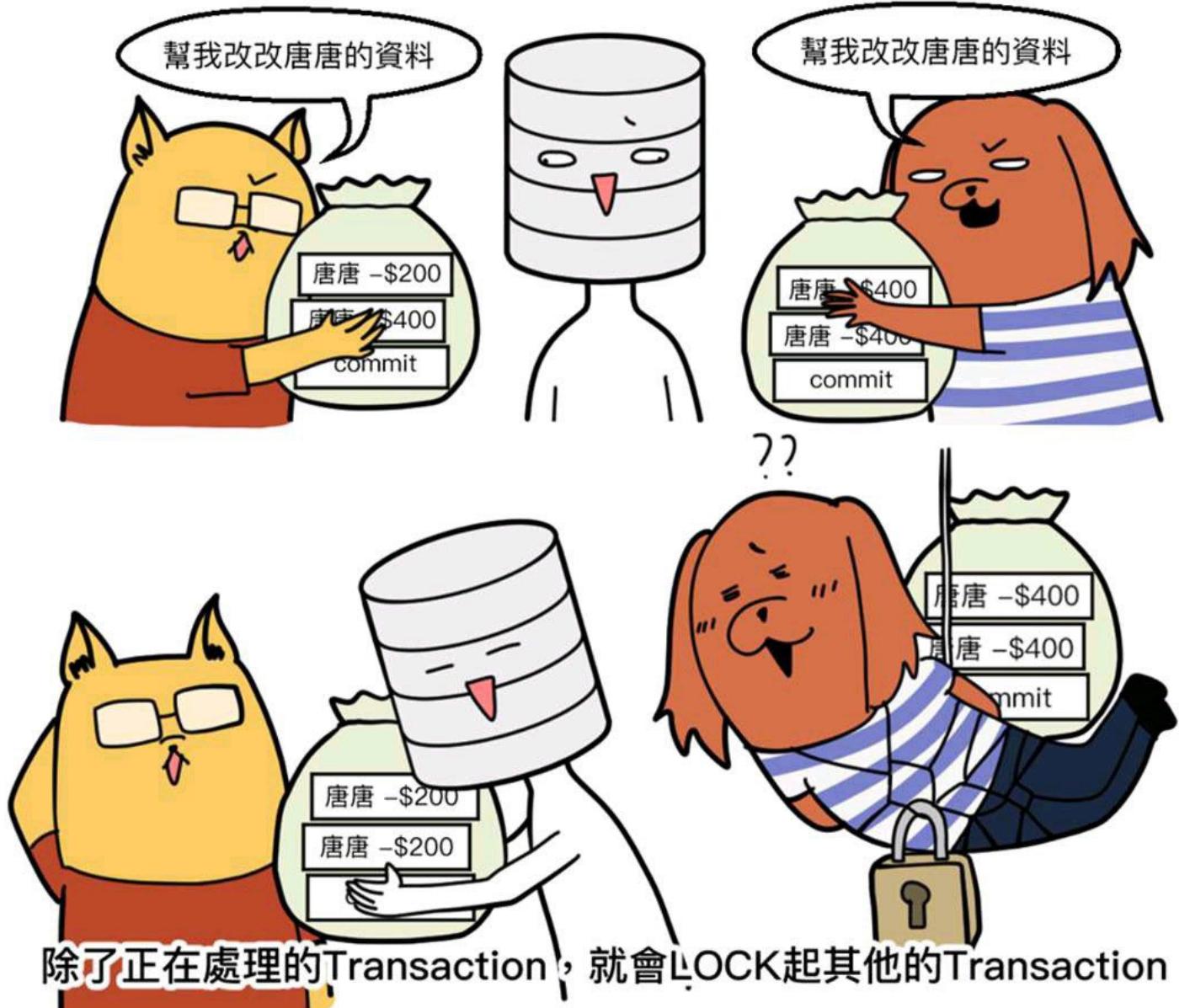
唐唐的結餘:
\$500

條件:
結餘不能為負數



Consistency ensures that a transaction can only bring the database from one valid state to another, maintaining database invariants: any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof. This prevents database corruption by an illegal transaction, but does not guarantee that a transaction is correct.

Isolation就是為了防止多個事務並發執行時
由於交叉執行而導致數據的不一致。



Transactions are often executed concurrently (e.g., reading and writing to multiple tables at the same time). Isolation ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially. Isolation is the main goal of concurrency control; depending on the method used, the effects of an incomplete transaction might not even be visible to other transactions.

Durability就是保證Commit後資料不會流失



除非是硬件問題
例如Harddisk data corruption

如果喜歡IT狗教學幼幼文
請按讚和分享~~~~

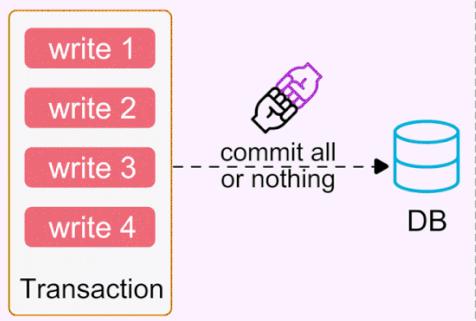


Durability guarantees that once a transaction has been committed, it will remain committed even in the case of a system failure (e.g., power outage or crash). This usually means that completed transactions (or their effects) are recorded in non-volatile memory.

What does ACID Mean?

Atomicity

All or nothing



Consistency

Preserving database invariants

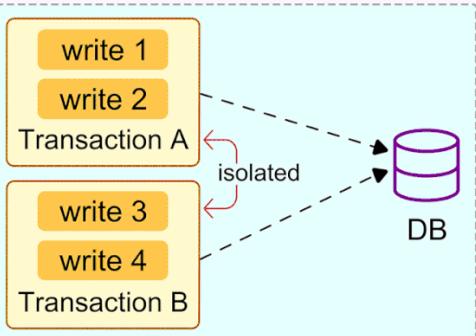
consistent state A

Transactions

consistent state B

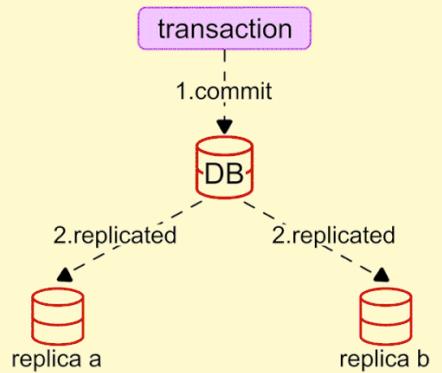
Isolation

Concurrent transactions are isolated from each other



Durability

Data is persisted after transaction is committed even in a system failure



Deadlock



COP: Thread #1 demands Resource #2 but Criminal owns the LOCK

CRIMINAL: Thread #2 demands Resource #1 but Cop owns the LOCK

CRIMINALS FRIEND: Resource #2, the owner of the LOCK is Cop

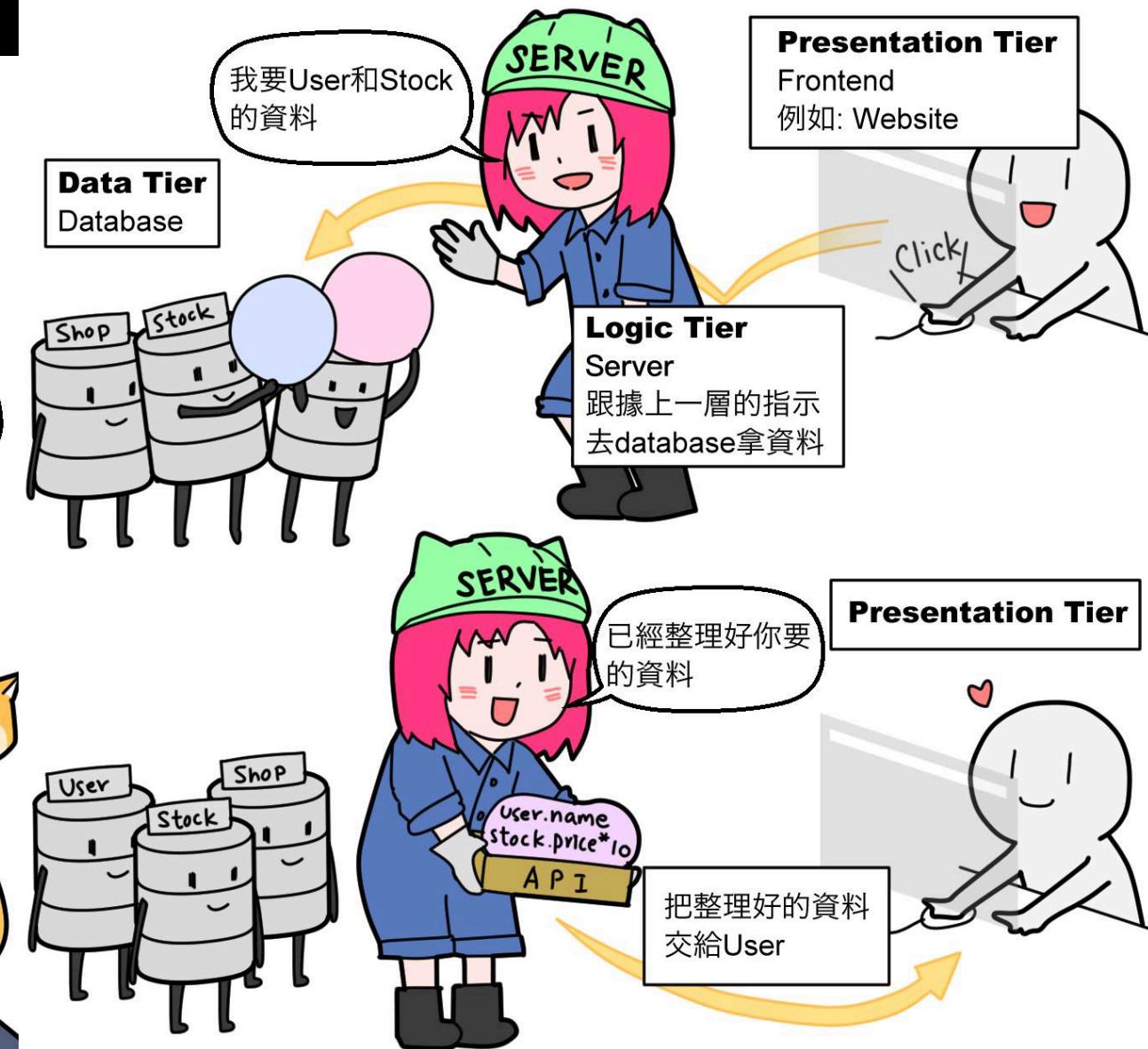
HOSTAGE OF CRIMINAL: Resource #1, the owner of the LOCK is CRIMINAL

Two or more entities are blocking some sources, and none of them is able to finish, because their are blocking sources in a cyclic way.

Step	Transaction T_1	Transaction T_2
1	UPDATE authors SET citations=100 WHERE paperid=1	
2		UPDATE titles SET copyright=1 WHERE titleid=2
3	SELECT title, doi FROM titles WHERE titleid=2	
4		SELECT authorname FROM authors WHERE paperid=1

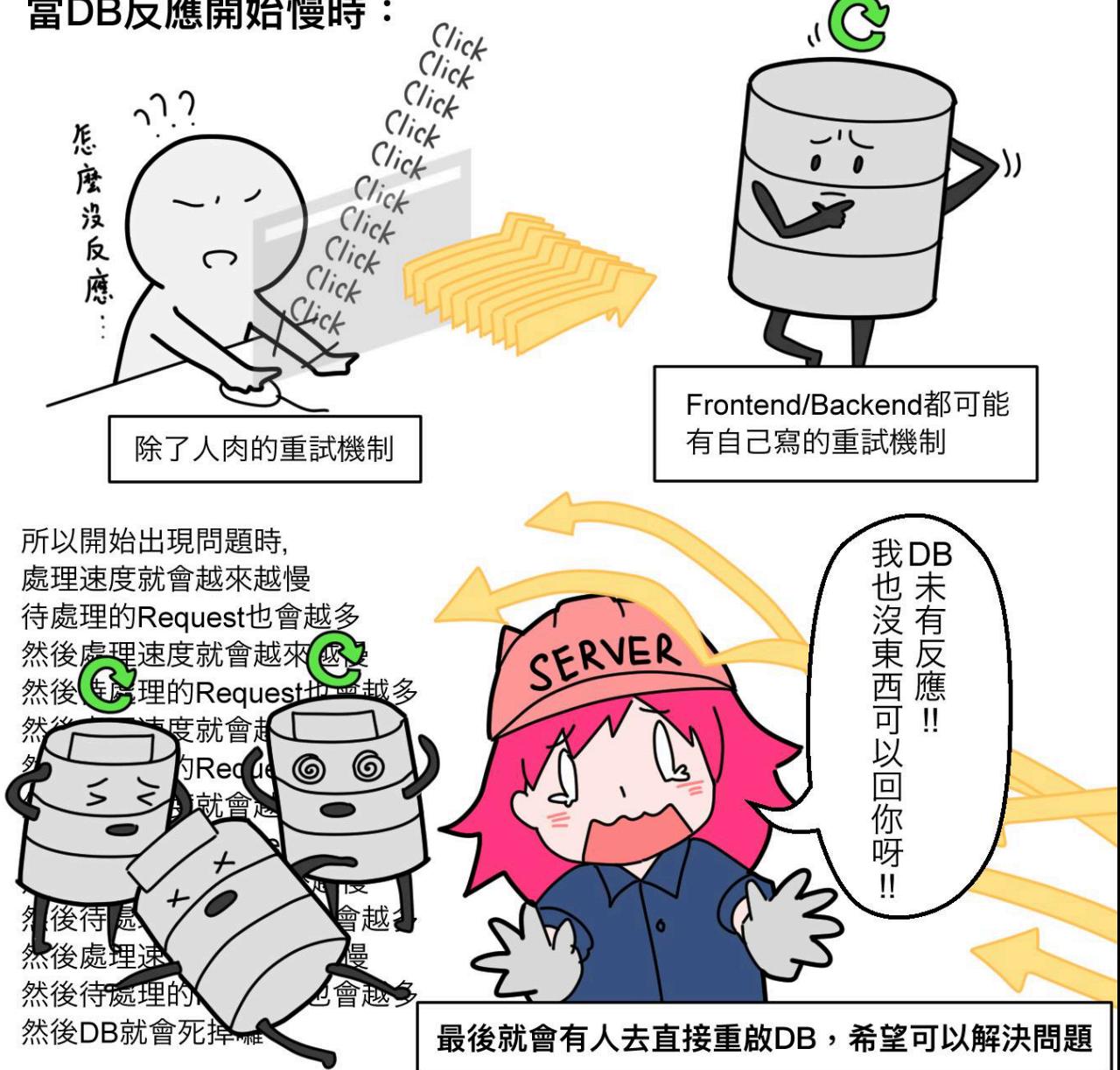
什麼時候Database的負載是最高的？

~高流量處理小心得~

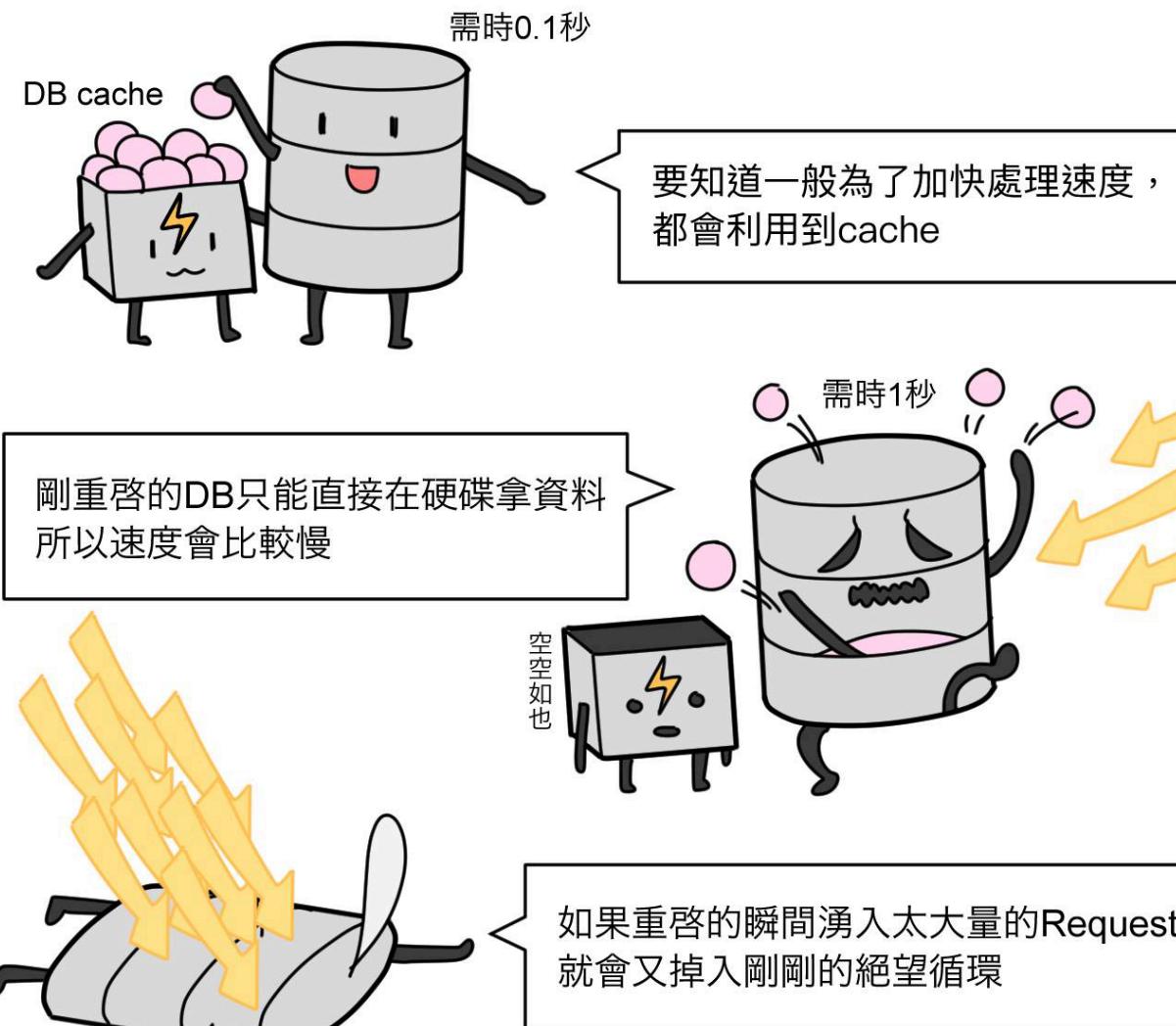


但直接重啟DB後，為什麼馬上又會當機？

當DB反應開始慢時：



最後就會有人去直接重啟DB，希望可以解決問題



剛重啟DB時，可以有限量地回復DB的服務(throttling)
給予DB足够的Warm-Up Time

要走出絕望循環，做法可以是先閹掉或轉移
來自外面的Request

