

The Great Unknown. All About NULL in the Oracle database.

This is the size of the smallest text in the presentation, if you can't read this move closer!

I'm English, I'd rather drink tea than talk about myself, so let's get that out way here...

Name: Patrick Jolliffe (say it like it rhymes with shoplift. Two Ls, two Fs, one E)

Twitter: @jolliffe (mostly over it, but should respond to mentions & DMs)

(Simplified) Location History: England (the South) -> Hong Kong (Island) -> Portugal (near Lisbon)

Work Status: (More than) Semi-retired Oracle DBA (not seeking work), Symposium 42 member, ACE Ass. ♠

Likes: Cinema (horror), video games (Monster Hunter), music (rock), manga (One Piece), dogs, Portugal.

Dislikes: Poor cinema etiquette, totalitarianist regimes, Brexit, racists, the French.

Disclaimer: No animals were harmed in the making of this presentation. Cats are animals, you say?

Revised Disclaimer: No dogs were harmed in the making of this presentation.

The Great Unknown. All About NULL in the Oracle database.

This is the size of the smallest text in the presentation, if you can't read this move closer!

I'm English, I'd rather drink tea than talk about myself, so let's get that out way here...

Name: Patrick Jolliffe (say it like it rhymes with shoplift. Two Ls, two Fs, one E)

Twitter: @jolliffe (mostly over it, but should respond to mentions & DMs)

(Simplified) Location History: England (the South) -> Hong Kong (Island) -> Portugal (near Lisbon)

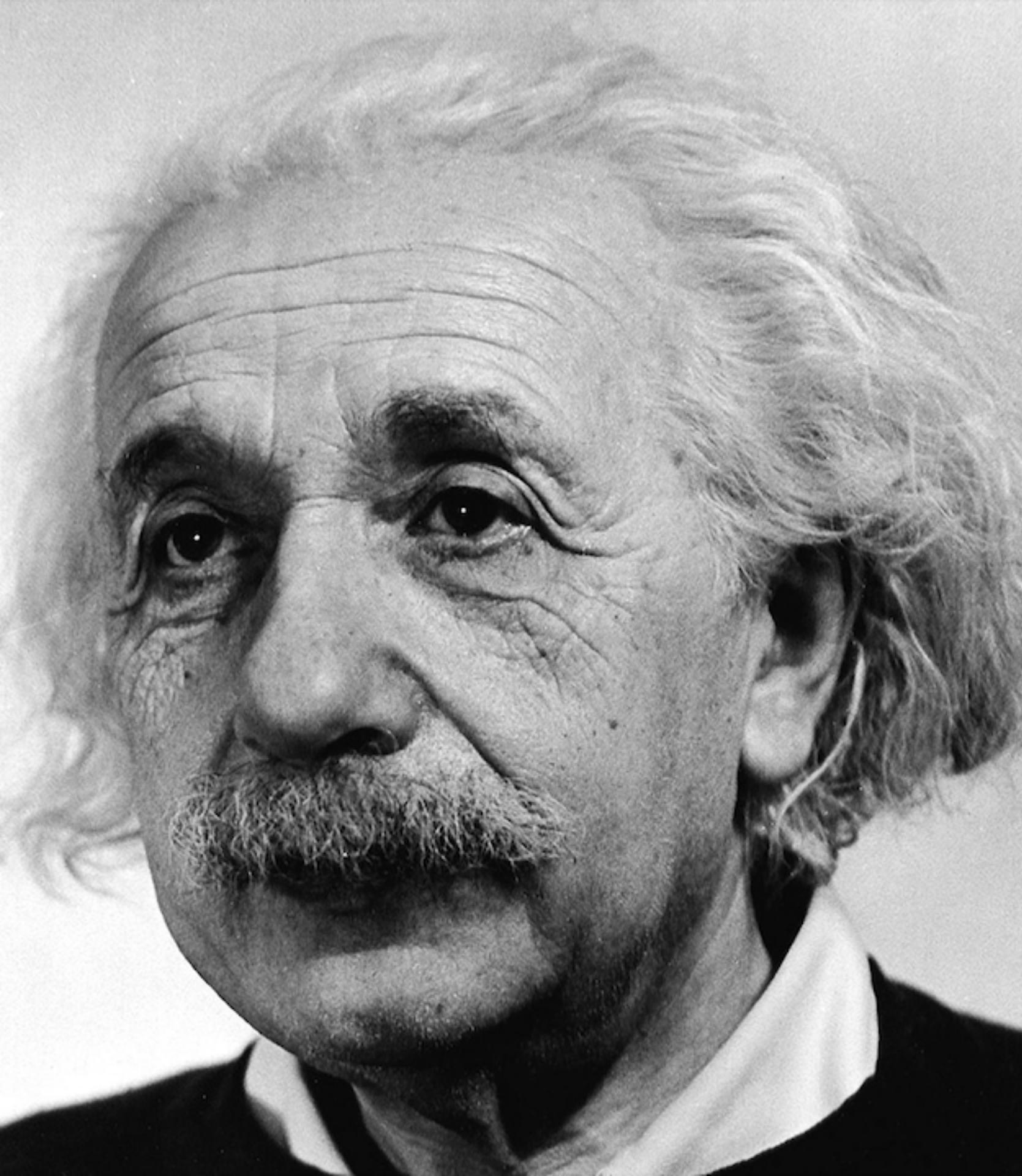
Work Status: (More than) Semi-retired Oracle DBA (not seeking work), Symposium 42 member, ACE Ass. ♠

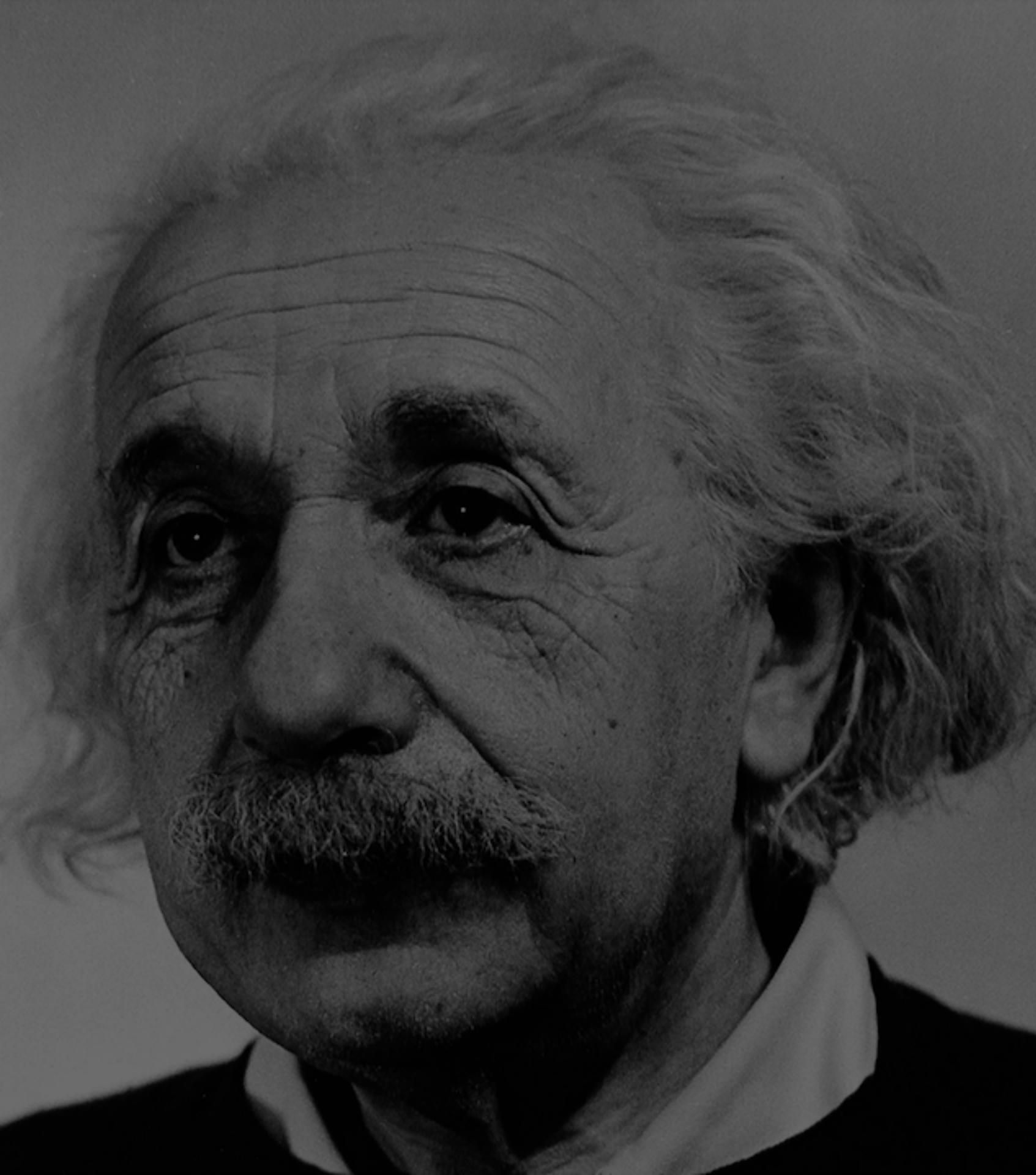
Likes: Cinema (horror), video games (Monster Hunter), music (rock), manga (One Piece), dogs, Portugal.

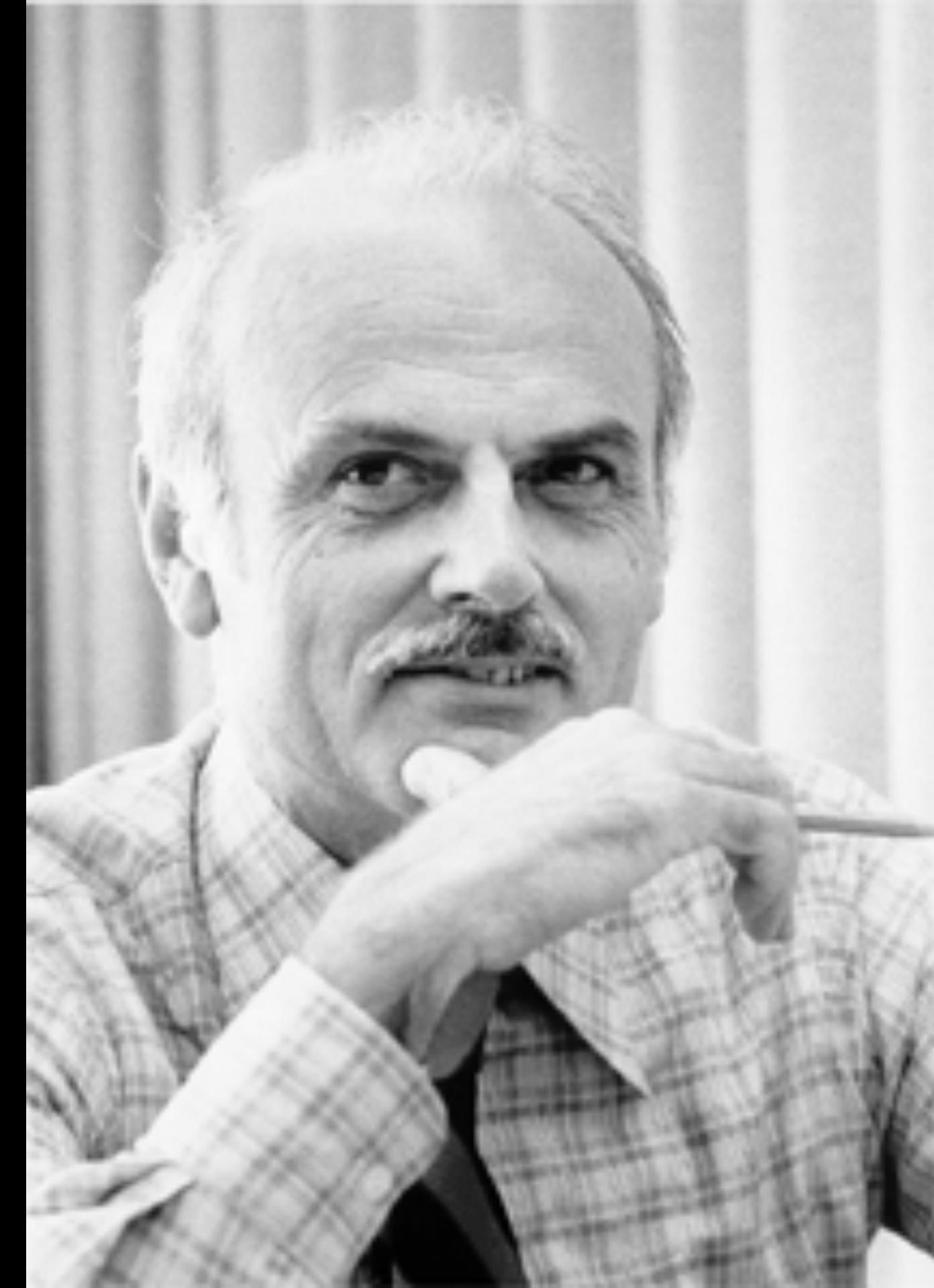
Dislikes: Poor cinema etiquette, totalitarianist regimes, Brexit, racists, the French.

Disclaimer: No animals were harmed in the making of this presentation. Cats are animals, you say?

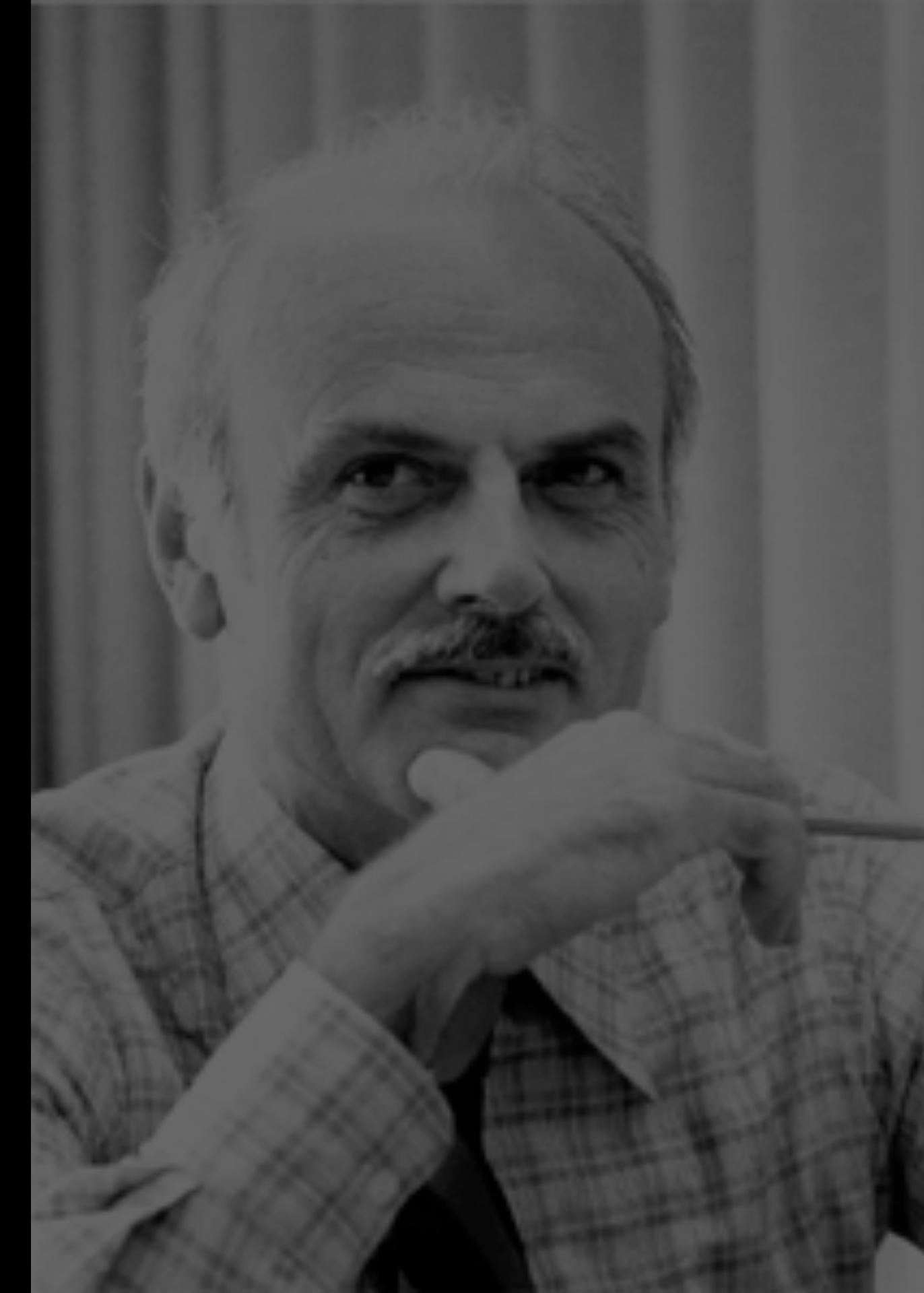
Revised Disclaimer: No dogs were harmed in the making of this presentation.







1970



Sun
1Mr
> Search program

Timezone ▾

Presentation Type ▾

Track ▾

Oracle Cloud Applications ▾

Oracle Technology ▾

GenAI ▾

Oracle Cloud Infrastructure ▾



Customer story ▾

Oracle Applications Unlimited ▾

Community ▾

Your bookmarks ▾



Belmont 1 Ground Floor

Belmont 2 Ground Floor

Belmont 3 Ground Floor

Belmont 4 Ground Floor

④ Belmont 1 Ground Floor 12:00-13:00

**Oracle ADB Security - Beyond the Standard Configurations**

Presentation type: Live Demo

Track: Oracle Cloud Infrastructure

[1 submission](#)

④ Belmont 2 Ground Floor 12:00-13:00

**APEX, ORDS and JRDV - The Three Amigos**

Presentation type: Presentation (45 minutes)

Track: Oracle Technology

[1 submission](#)

④ Belmont 3 Ground Floor 12:00-13:00

**Upgrade to Oracle Database 23ai: Best Practices and Customer Experience**

Presentation type: Presentation (45 minutes)

Track: Oracle Technology

[1 submission](#)

④ Belmont 4 Ground Floor 12:00-13:00

**Effective Data Modeling with 23ai JSON Duality Views and JSON Collections**

Presentation type: Live Demo

Track: Oracle Technology

[1 submission](#)

13:00

④ Belmont 1 Ground Floor 13:00-14:00

**ORDS in depth**

Presentation type: Presentation (45 minutes)

Track: Oracle Technology

[1 submission](#)

④ Belmont 2 Ground Floor 13:00-14:00

**When "low code" becomes "low level code". A.K.A. Dtrace and GDB for beginners.**

Presentation type: Live Demo

Track: Oracle Technology

[1 submission](#)

④ Belmont 3 Ground Floor 13:00-14:00

**Oracle GoldenGate 23ai: Transitioning from Classic to Microservices Architecture**

Presentation type: Presentation (45 minutes)

Track: Oracle Technology

[1 submission](#)

④ Belmont 4 Ground Floor 13:00-14:00

**Migrations from MongoDB, Couchbase, DocumentDB, etc to Oracle Autonomous Database**

Presentation type: Live Demo

Track: Oracle Technology

[1 submission](#)

14:00

④ Belmont 1 Ground Floor 14:00-15:00

**Sponsor**

Presentation type: Presentation (45 minutes)

Track: Oracle Technology

[1 submission](#)

④ Belmont 2 Ground Floor 14:00-15:00

**Migrating the Beast—Transportable Tablespaces to the Extreme**

Presentation type: Presentation (45 minutes)

Track: Oracle Technology

[1 submission](#)

④ Belmont 3 Ground Floor 14:00-15:00

**Oracle Database 23ai for Developers**

Presentation type: Live Demo

Track: Oracle Technology

[1 submission](#)

④ Belmont 4 Ground Floor 14:00-15:00

**Sponsor**

Track: Oracle Cloud Infrastructure

15:00

④ Belmont 1 Ground Floor 15:00-16:00

**Application Continuity with Oracle Database 23ai**

Presentation type: Presentation (45 minutes)

Track: Oracle Technology

[1 submission](#)

④ Belmont 2 Ground Floor 15:00-16:00

**Blockbuster! A Guide to Oracle Block Internals, and Hacking Them**

Presentation type: Presentation (45 minutes)

Track: Oracle Technology

[1 submission](#)

④ Belmont 3 Ground Floor 15:00-16:00

**Monitoring GoldenGate MicroServices - A New Approach**

Presentation type: Presentation (45 minutes)

Track: Oracle Technology

[1 submission](#)

④ Belmont 4 Ground Floor 15:00-16:00

**APEX Plugins - How to get started**

Presentation type: Hands-on-Lab

Track: Oracle Technology

[1 submission](#)

16:00

speakers	name	handle	style
	Jonathan	@jloracle	
	Patrick	@jolliffe	
sessions	speaker	title	attendees
	Jonathan	Analytical Asparagus	90
	Jonathan	Sourdough Schemas	9
	Patrick	Dogs vs Cats	0

speakers	name	handle	style
	Jonathan	@jloracle	
	Patrick	@jolliffe	
sessions	speaker	title	attendees
	Jonathan	Analytical Asparagus	90
	Jonathan	Sourdough Schemas	9
	Patrick	Dogs vs Cats	0
	Patrick	NULL	

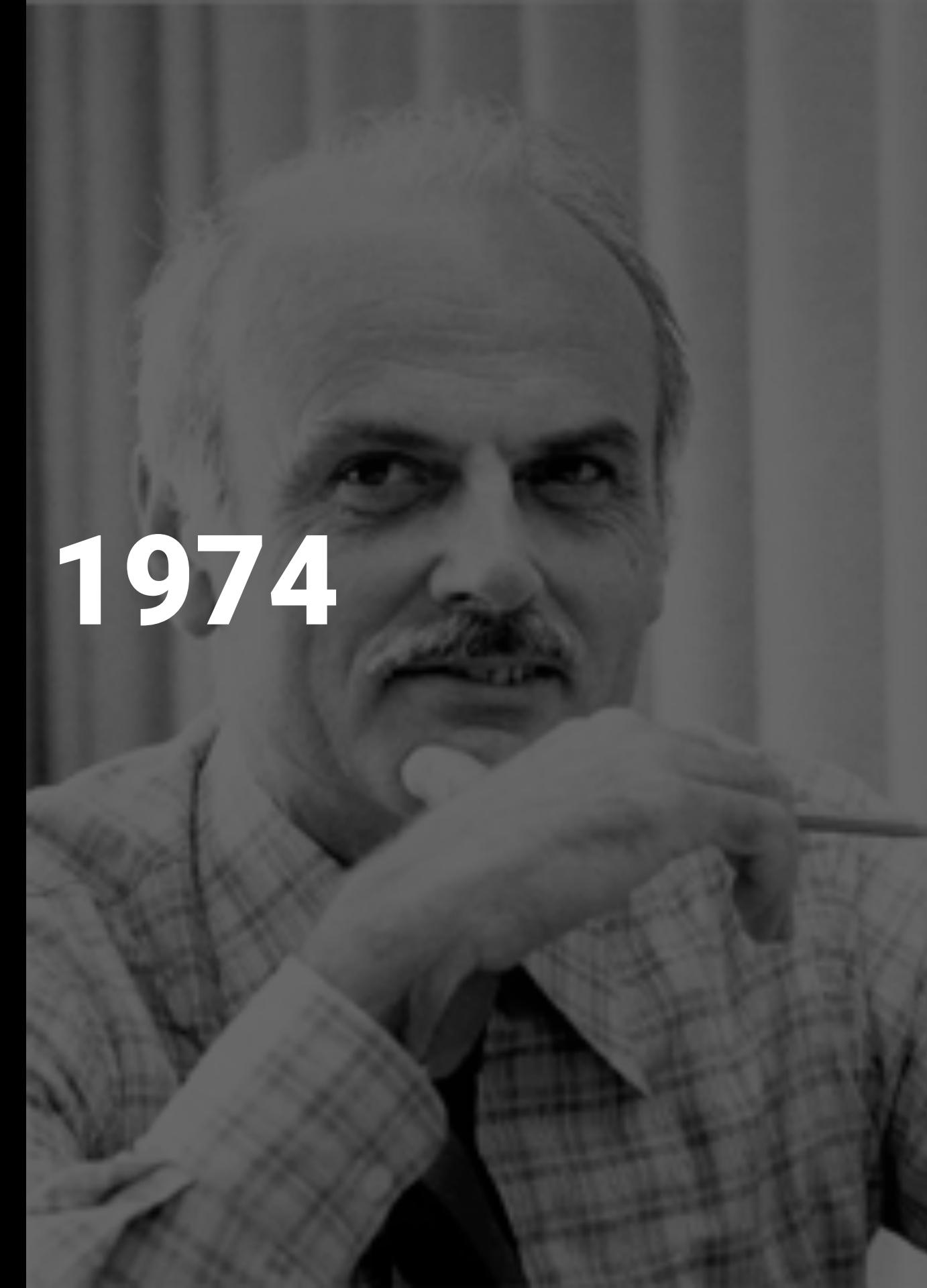
speakers	name	handle	style
	Jonathan	@jloracle	
	Patrick	@jolliffe	
sessions	speaker	title	attendees
	Jonathan	Analytical Asparagus	90
	Jonathan	Sourdough Schemas	9
	Patrick	Dogs vs Cats	0
	Patrick	NULL	NULL

speakers	name	handle	style
	Jonathan	@jloracle	
	Patrick	@jolliffe	
sessions	speaker	title	attendees
	Jonathan	Analytical Asparagus	90
	Jonathan	Sourdough Schemas	9
	Patrick	Dogs vs Cats	0
	Patrick	NULL	0

speakers	name	handle	style
	Jonathan	@jloracle	
	Patrick	@jolliffe	
sessions	speaker	title	attendees
	Jonathan	Analytical Asparagus	90
	Jonathan	Sourdough Schemas	9
	Patrick	Dogs vs Cats	0
	Patrick	NULL	-99

1970

1974

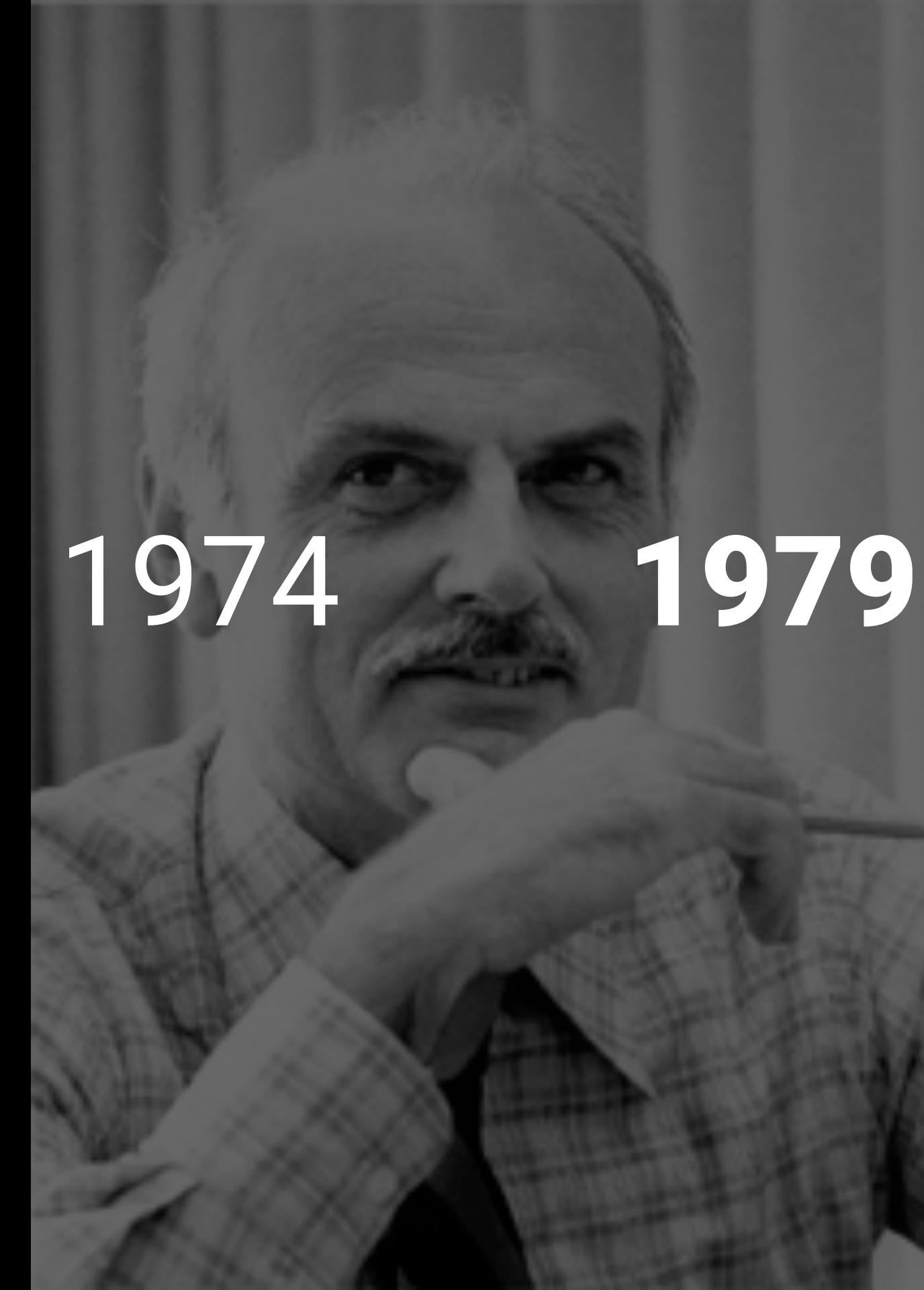


speakers	name	handle	style
	Jonathan	@jloracle	
	Patrick	@jolliffe	
sessions	speaker	title	attendees
	Jonathan	Analytical Asparagus	90
	Jonathan	Sourdough Schemas	9
	Patrick	Dogs vs Cats	0
	Patrick	NULL	[null]

1970

1974

1979



special marker

special marker

value does not exist

oxymoron

crash landing

paid volunteer

herbal tea

paper towel

minor miracle

plastic glass

null value

random order

old technology

small crowd

Boolean Logic

Three-Valued Logic

Expression

Result

false and null

false

true or null

true

Expression

Result

not null

true and null

null and null

false or null

null or null





WANTED



DEAD & ALIVE
SCHRÖDINGER'S CAT

```
> create table boxes(id      number null,  
2*                  live_cats number null);
```

Table BOXES created.



```
> insert into boxes (id, live_cats)  
2          values ( 1,      1),  
3*        ( 2,      0);
```

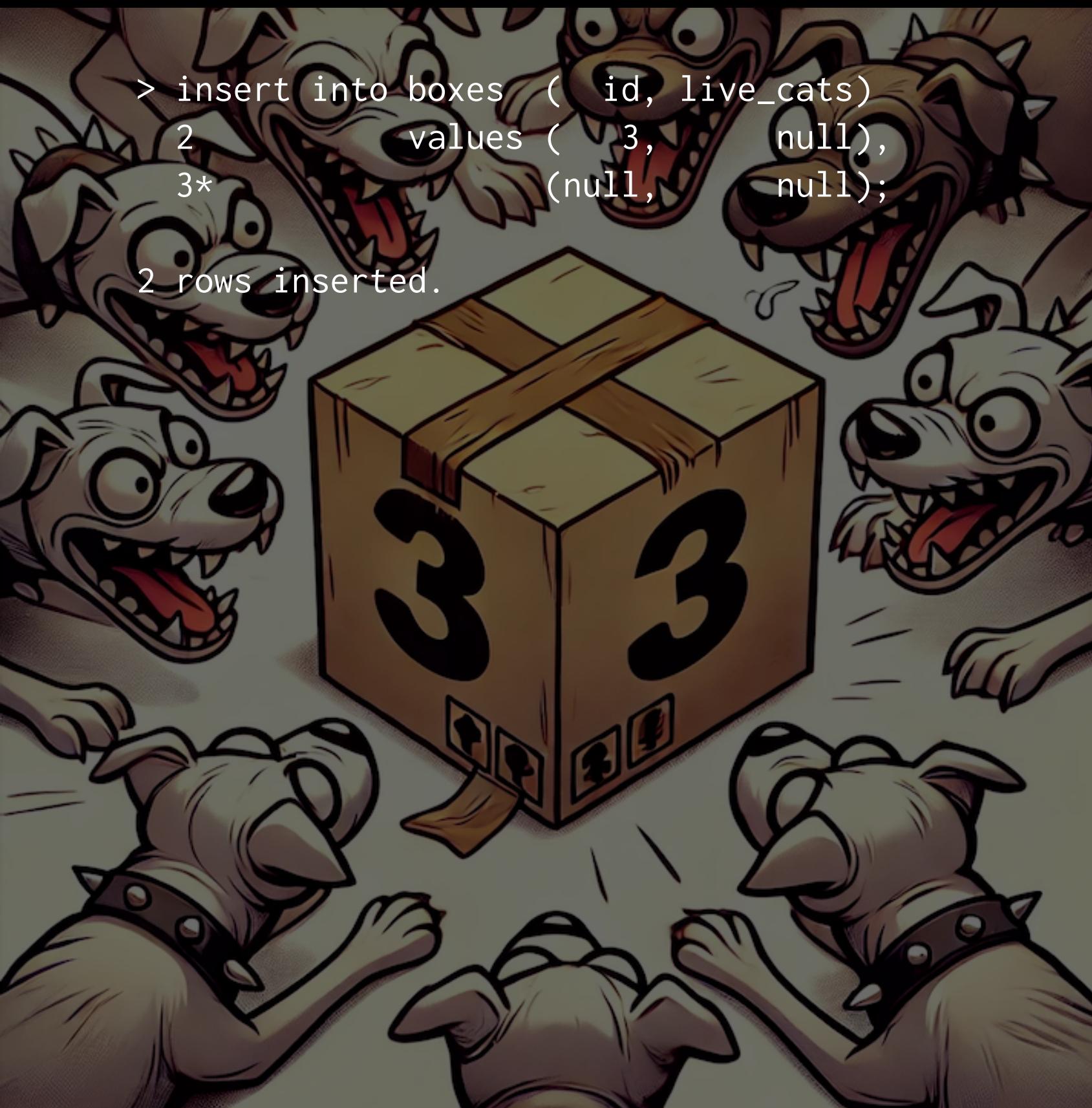
2 rows inserted.





```
> insert into boxes ( id, live_cats)  
2      values ( 3,      null),  
3*    (null,      null);
```

2 rows inserted.



```
> select * from boxes;
```

ID	LIVE_CATS
1	1
2	0
3	

```
> set null [null]  
> select * from boxes;
```

ID	LIVE_CATS
1	1
2	0
3	[null]
[null]	[null]

```
> select * from boxes;
```

ID	LIVE_CATS
1	1
2	0
3	

```
> set null [null]  
> select * from boxes;
```

ID	LIVE_CATS
1	1
2	0
3	[null]
[null]	[null]

```
> select * from boxes;
```

ID	LIVE_CATS
1	1
2	0
3	

```
> set null [null]  
> select * from boxes;
```

ID	LIVE_CATS
1	1
2	0
3	[null]
[null]	[null]



CATQUERIES.COM
THE PLACE FOR ALL YOUR CAT QUERIES

```
> select id, live_cats from boxes where (live_cats is not null);
```

ID	LIVE_CATS
1	1
2	0

```
> select id, live_cats from boxes where (live_cats is null);
```

ID	LIVE_CATS
3	[null]
[null]	[null]

```
> select id, live_cats from boxes where (live_cats is not null);
```

ID	LIVE_CATS
1	1
2	0

```
> select id, live_cats from boxes where (live_cats is null);
```

ID	LIVE_CATS
3	[null]
[null]	[null]

```
> select * from boxes where null;
```

```
no rows selected
```

```
> select id, live_cats, (live_cats = live_cats), (live_cats != live_cats) from boxes;
```

ID	LIVE_CATS	(LIVE_CATS = LIVE_CATS)	(LIVE_CATS != LIVE_CATS)
1	1	TRUE	FALSE
2	0	TRUE	FALSE
3	[null]	[null]	[null]
[null]	[null]	[null]	[null]

```
> select id, live_cats from boxes where (live_cats = live_cats);
```

ID	LIVE_CATS
1	1
2	0

```
> select id, live_cats, (live_cats = live_cats), (live_cats != live_cats) from boxes;
```

ID	LIVE_CATS	(LIVE_CATS = LIVE_CATS)	(LIVE_CATS != LIVE_CATS)
1	1	TRUE	FALSE
2	0	TRUE	FALSE
3	[null]	[null]	[null]
[null]	[null]	[null]	[null]

```
> select id, live_cats from boxes where (live_cats = live_cats);
```

ID	LIVE_CATS
1	1
2	0

```
> select id, live_cats, (live_cats = live_cats), (live_cats != live_cats) from boxes;
```

ID	LIVE_CATS	(LIVE_CATS = LIVE_CATS)	(LIVE_CATS != LIVE_CATS)
1	1	TRUE	FALSE
2	0	TRUE	FALSE
3	[null]	[null]	[null]
[null]	[null]	[null]	[null]

```
> select id, live_cats from boxes where (live_cats = live_cats);
```

ID	LIVE_CATS
1	1
2	0

```
> select id, live_cats, (live_cats = 0), (live_cats != 0) from boxes;
```

ID	LIVE_CATS	(LIVE_CATS = 0)	(LIVE_CATS != 0)
1	1	FALSE	TRUE
2	0	TRUE	FALSE
3	[null]	[null]	[null]
[null]	[null]	[null]	[null]

```
> select count(*) from boxes where (live_cats = 0) or (live_cats != 0);
```

COUNT(*)
2

```
> select count(*) from boxes where (live_cats = 0) or (live_cats != 0) or (live_cats is null);
```

COUNT(*)
4

```
> select id, live_cats, (live_cats = 0), (live_cats != 0) from boxes;
```

ID	LIVE_CATS	(LIVE_CATS = 0)	(LIVE_CATS != 0)
1	1	FALSE	TRUE
2	0	TRUE	FALSE
3	[null]	[null]	[null]
[null]	[null]	[null]	[null]

```
> select count(*) from boxes where (live_cats = 0) or (live_cats != 0);
```

COUNT(*)
2

```
> select count(*) from boxes where (live_cats = 0) or (live_cats != 0) or (live_cats is null);
```

COUNT(*)
4

```
> select id, live_cats, (live_cats = 0), (live_cats != 0) from boxes;
```

ID	LIVE_CATS	(LIVE_CATS = 0)	(LIVE_CATS != 0)
1	1	FALSE	TRUE
2	0	TRUE	FALSE
3	[null]	[null]	[null]
[null]	[null]	[null]	[null]

```
> select count(*) from boxes where (live_cats = 0) or (live_cats != 0);
```

COUNT(*)
2

```
> select count(*) from boxes where (live_cats = 0) or (live_cats != 0) or (live_cats is null);
```

COUNT(*)
4

```
> select id, live_cats, (live_cats = 0), (live_cats != 0) from boxes;
```

ID	LIVE_CATS	(LIVE_CATS = 0)	(LIVE_CATS != 0)
1	1	FALSE	TRUE
2	0	TRUE	FALSE
3	[null]	[null]	[null]
[null]	[null]	[null]	[null]

```
> select count(*) from boxes where (live_cats = 0) or (live_cats != 0);
```

COUNT(*)
2

```
> select count(*) from boxes where (live_cats = 0) or (live_cats != 0) or (live_cats is null);
```

COUNT(*)
4

Operations

```
> select (null*42), (null/42), (null+42), (null-42);
```

(NULL*42)	(NULL/42)	(NULL+42)	(NULL-42)
[null]	[null]	[null]	[null]

```
> select (null*0), (null/0), (null/null), (null-null);
```

(NULL*0)	(NULL/0)	(NULL/NUL)	(NULL-NUL)
[null]	[null]	[null]	[null]

```
> select (null*42), (null/42), (null+42), (null-42);
```

(NULL*42)	(NULL/42)	(NULL+42)	(NULL-42)
-----	-----	-----	-----
[null]	[null]	[null]	[null]

```
> select (null*0), (null/0), (null/null), (null-null);
```

(NULL*0)	(NULL/0)	(NULL/NUL)	(NULL-NUL)
-----	-----	-----	-----
[null]	[null]	[null]	[null]



```
> select 'cock' || null || 'womble';
```

cockwomble

```
> select 'cock' || null || 'womble';
```

```
cockwomble
```



Aggregate Functions

```
> select id, live_cats from boxes;
```

ID	LIVE_CATS
1	1
2	0
3	[null]
[null]	[null]

```
> select count(null), count(live_cats), count(*) from boxes;
```

COUNT(NULL)	COUNT(LIVE_CATS)	COUNT(*)
0	2	4

```
> select id, live_cats from boxes;
```

ID	LIVE_CATS
1	1
2	0
3	[null]
[null]	[null]

```
> select count(null), count(live_cats), count(*) from boxes;
```

COUNT(NULL)	COUNT(LIVE_CATS)	COUNT(*)
0	2	4

```
> select live_cats from boxes;
```

LIVE_CATS
1
0
[null]
[null]

```
> select sum(null), sum(live_cats) from boxes;
```

SUM(NULL)	SUM(LIVE_CATS)
-----------	----------------

[null]	1
--------	---

```
> select live_cats from boxes;
```

LIVE_CATS

1

0

[null]

[null]

```
> select avg(null), avg(live_cats) from boxes;
```

AVG(NULL) AVG(LIVE_CATS)

[null]

0.5

```
> select live_cats from boxes;
```

LIVE_CATS
1
0
[null]
[null]

```
> select avg(null), avg(live_cats) from boxes;
```

AVG(NULL)	AVG(LIVE_CATS)
[null]	0.5



```
> select live_cats from boxes;
```

LIVE_CATS
1
0
[null]
[null]

```
> select min(null), max(null), min(live_cats), max(live_cats) from boxes;
```

MIN(NULL)	MAX(NULL)	MIN(LIVE_CATS)	MAX(LIVE_CATS)
-----------	-----------	----------------	----------------

[null]

0



group by & distinct

```
> select live_cats, count(*) from boxes group by live_cats;
```

LIVE_CATS	COUNT(*)
1	1
0	1
[null]	2

```
> select distinct live_cats from boxes;
```

LIVE_CATS
1
0
[null]

```
> select live_cats, count(*) from boxes group by live_cats;
```

LIVE_CATS	COUNT(*)
1	1
0	1
[null]	2

```
> select distinct live_cats from boxes;
```

LIVE_CATS
1
0
[null]



Shhhh! I ordering noms.

A fluffy, light-colored cat with dark stripes is sitting on a dark surface, looking intently at a smartphone screen. The phone is held vertically, showing a small image of the same cat. The background is a dark, reddish-brown color.

ordering

Shhhh! I ordering noms.

```
> select live_cats from boxes order by live_cats;
```

LIVE_CATS

0

1

[null]

[null]

```
> select live_cats from boxes order by live_cats nulls first;
```

LIVE_CATS

[null]

[null]

0

1

```
> select live_cats from boxes order by live_cats;
```

LIVE_CATS

0

1

[null]

[null]

```
> select live_cats from boxes order by live_cats nulls first;
```

LIVE_CATS

[null]

[null]

0

1

```
> select live_cats from boxes order by live_cats desc;
```

LIVE_CATS

```
-----  
[null]  
[null]  
1  
0
```

```
> select live_cats from boxes order by live_cats desc nulls last;
```

LIVE_CATS

```
-----  
1  
0  
[null]  
[null]
```

```
> select live_cats from boxes order by live_cats desc;
```

LIVE_CATS

```
-----  
[null]  
[null]  
1  
0
```

```
> select live_cats from boxes order by live_cats desc nulls last;
```

LIVE_CATS

```
-----  
1  
0  
[null]  
[null]
```



```
> select live_cats, nvl(live_cats, 0), coalesce(live_cats, 0) from boxes;
```

LIVE_CATS	NVL(LIVE_CATS,0)	COALESCE(LIVE_CATS,0)
1	1	1
0	0	0
[null]	0	0
[null]	0	0

```
> select live_cats, id, coalesce(live_cats, id, 0) from boxes;
```

LIVE_CATS	ID	COALESCE(LIVE_CATS, ID, 0)
1	1	1
0	2	0
[null]	3	3
[null]	[null]	0

```
> select live_cats, nvl(live_cats, 0), coalesce(live_cats, 0) from boxes;
```

LIVE_CATS	NVL(LIVE_CATS,0)	COALESCE(LIVE_CATS,0)
1	1	1
0	0	0
[null]	0	0
[null]	0	0

```
> select live_cats, id, coalesce(live_cats, id, 0) from boxes;
```

LIVE_CATS	ID	COALESCE(LIVE_CATS, ID, 0)
1	1	1
0	2	0
[null]	3	3
[null]	[null]	0

```
> select coalesce(1);
```

SQL Error: ORA-00938: not enough arguments for function

*Cause: The function was referenced with too few arguments.

```
> select coalesce(
```

```
2      1,      2,      3,      4,      5,      6,      7,      8,      9,      10,
```

```
3      11,     12,     13,     14,     15,     16,     17,     18,     19,     20,
```

```
4      21,     22,     23,     24,     25,     26,     27,     28,     29,     30,
```

```
5      31,     32,     33,     34,     35,     36,     37,     38,     39,     40,
```

```
6      41,     42,     43,     44,     45,     46,     47,     48,     49,     50,
```

```
...
```

```
6552    65501, 65502, 65503, 65504, 65505, 65506, 65507, 65508, 65509, 65510,
```

```
6553    65511, 65512, 65513, 65514, 65515, 65516, 65517, 65518, 65519, 65520,
```

```
6554    65521, 65522, 65523, 65524, 65525, 65526, 65527, 65528, 65529, 65530,
```

```
6555*   65531, 65532, 65533, 65534, 65535, 65536);
```

ERROR at line 3:

ORA-00939: too many arguments for function

```
> select coalesce(1);
```

SQL Error: ORA-00938: not enough arguments for function

*Cause: The function was referenced with too few arguments.

```
> select coalesce(
```

```
2      1,      2,      3,      4,      5,      6,      7,      8,      9,      10,
```

```
3      11,     12,     13,     14,     15,     16,     17,     18,     19,     20,
```

```
4      21,     22,     23,     24,     25,     26,     27,     28,     29,     30,
```

```
5      31,     32,     33,     34,     35,     36,     37,     38,     39,     40,
```

```
6      41,     42,     43,     44,     45,     46,     47,     48,     49,     50,
```

...

```
6552    65501, 65502, 65503, 65504, 65505, 65506, 65507, 65508, 65509, 65510,
```

```
6553    65511, 65512, 65513, 65514, 65515, 65516, 65517, 65518, 65519, 65520,
```

```
6554    65521, 65522, 65523, 65524, 65525, 65526, 65527, 65528, 65529, 65530,
```

```
6555*   65531, 65532, 65533, 65534, 65535, 65536);
```

ERROR at line 3:

ORA-00939: too many arguments for function

NVL**COALESCE**

SQL Standard



Number of Args

2

2->65,535

Oracle Version

All

>= 9i

Easy To Type



```
> select id from boxes where id = coalesce(:id, id);
```

Id	Operation	Name
0	SELECT STATEMENT	
* 1	INDEX FULL SCAN	BOX_ID

```
> select id from boxes where id = nvl(:id, id)
```

Id	Operation	Name
0	SELECT STATEMENT	
1	VIEW	VW_ORE_2032B19C
2	UNION-ALL	
* 3	FILTER	
* 4	INDEX RANGE SCAN	BOX_ID
* 5	FILTER	
* 6	INDEX FULL SCAN	BOX_ID

```
> select id from boxes where id = coalesce(:id, id);
```

Id	Operation	Name
0	SELECT STATEMENT	
* 1	INDEX FULL SCAN	BOX_ID

```
> select id from boxes where id = nvl(:id, id)
```

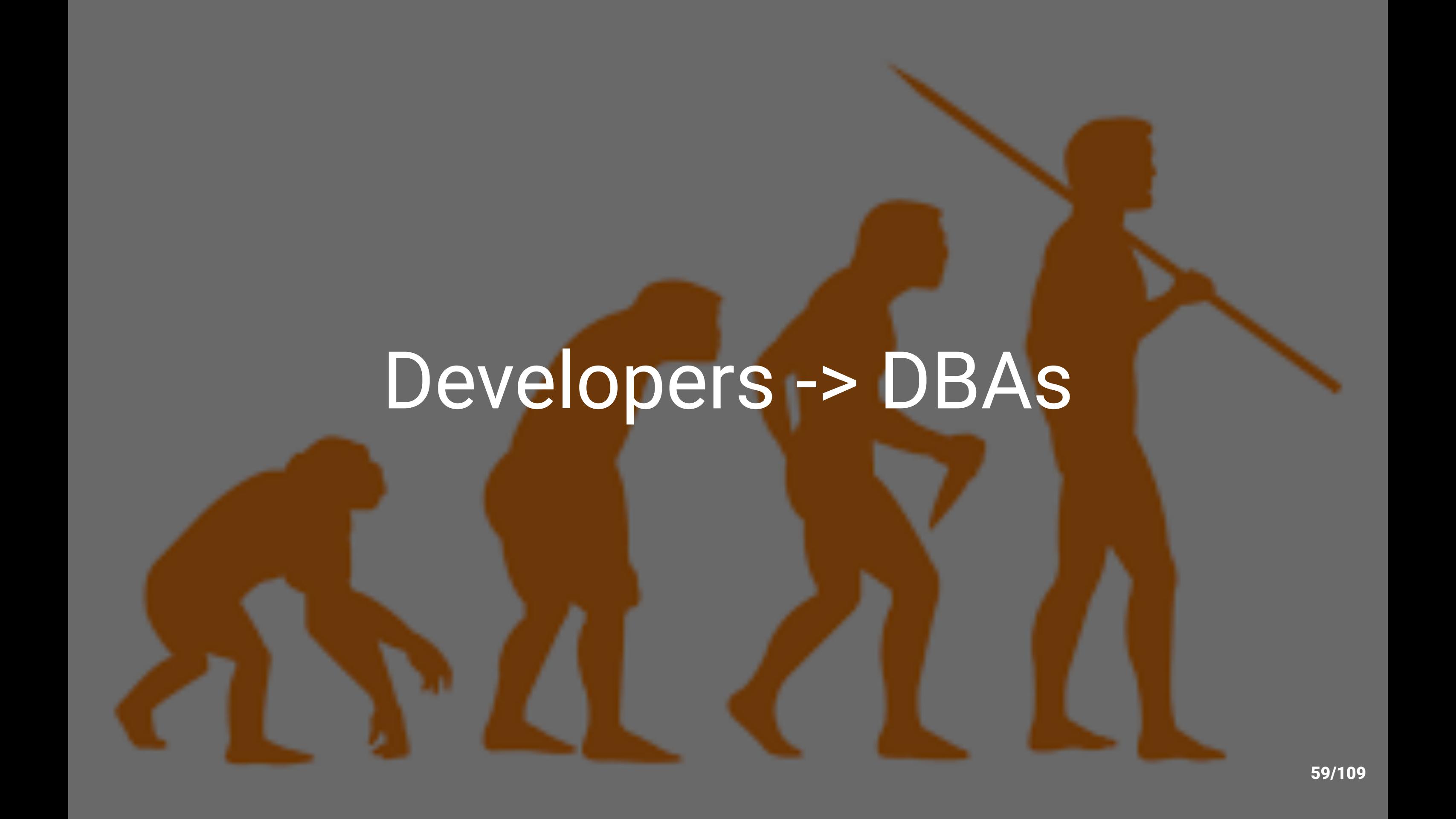
Id	Operation	Name
0	SELECT STATEMENT	
1	VIEW	VW_ORE_2032B19C
2	UNION-ALL	
* 3	FILTER	
* 4	INDEX RANGE SCAN	BOX_ID
* 5	FILTER	
* 6	INDEX FULL SCAN	BOX_ID

```
> select id from boxes where id = coalesce(:id, id);
```

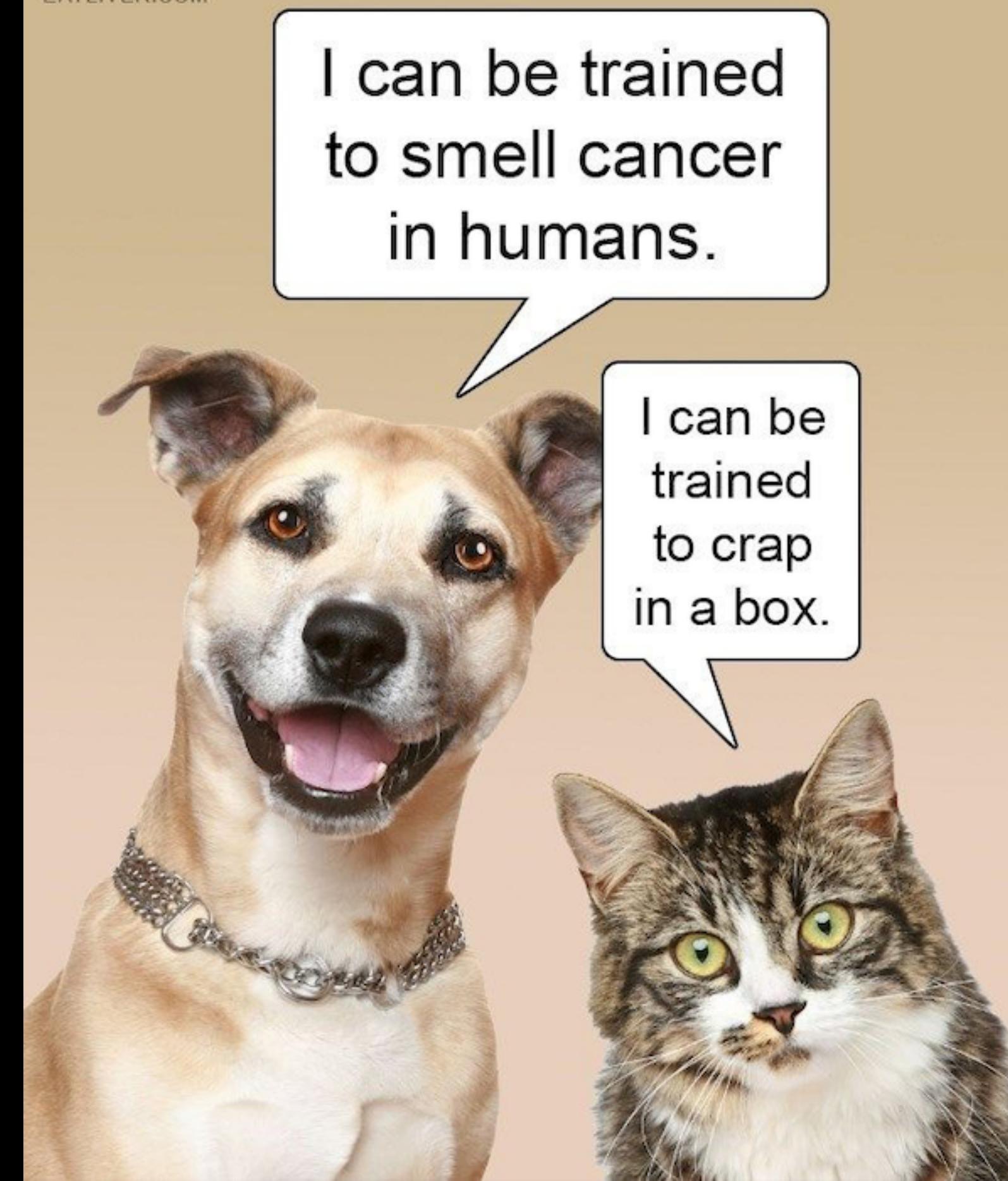
Id	Operation	Name
0	SELECT STATEMENT	
* 1	INDEX FULL SCAN	BOX_ID

```
> select id from boxes where id = nvl(:id, id)
```

Id	Operation	Name
0	SELECT STATEMENT	
1	VIEW	VW_ORE_2032B19C
2	UNION-ALL	
* 3	FILTER	
* 4	INDEX RANGE SCAN	BOX_ID
* 5	FILTER	
* 6	INDEX FULL SCAN	BOX_ID



Developers -> DBAs



Constraints

```
> create table pedigree_pooches(name varchar2(16),
2*                      breed varchar2(16) not null);
```

Table PEDIGREE_POOCHES created.

```
> select constraint_name, search_condition from user_constraints where table_name = 'PEDIGREE_POOCHES';
```

CONSTRAINT_NAME	SEARCH_CONDITION
SYS_C008494	"BREED" IS NOT NULL

```
> select column_name, nullable from user_tab_cols where table_name = 'PEDIGREE_POOCHES';
```

COLUMN_NAME	NULLABLE
NAME	Y
BREED	N

```
> insert into pedigree_pooches (name, breed) values ('Franck', null);
```

SQL Error: ORA-01400: cannot insert NULL into ("PDBADMIN"."PEDIGREE_POOCHES"."BREED")
*Cause: An attempt was made to insert NULL into previously listed objects.
*Action: These objects cannot accept NULL values. Reservable columns cannot accept NULL values.

```
> create table pedigree_pooches(name varchar2(16),
2*                      breed varchar2(16) not null);
```

Table PEDIGREE_POOCHES created.

```
> select constraint_name, search_condition from user_constraints where table_name = 'PEDIGREE_POOCHES';
```

CONSTRAINT_NAME	SEARCH_CONDITION
SYS_C008494	"BREED" IS NOT NULL

```
> select column_name, nullable from user_tab_cols where table_name = 'PEDIGREE_POOCHES';
```

COLUMN_NAME	NULLABLE
NAME	Y
BREED	N

```
> insert into pedigree_pooches (name, breed) values ('Franck', null);
```

SQL Error: ORA-01400: cannot insert NULL into ("PDBADMIN"."PEDIGREE_POOCHES"."BREED")
*Cause: An attempt was made to insert NULL into previously listed objects.
*Action: These objects cannot accept NULL values. Reservable columns cannot accept NULL values.

```
> create table pedigree_pooches(name varchar2(16),
2*                      breed varchar2(16) not null);
```

Table PEDIGREE_POOCHES created.

```
> select constraint_name, search_condition from user_constraints where table_name = 'PEDIGREE_POOCHES';
```

CONSTRAINT_NAME	SEARCH_CONDITION
SYS_C008494	"BREED" IS NOT NULL

```
> select column_name, nullable from user_tab_cols where table_name = 'PEDIGREE_POOCHES';
```

COLUMN_NAME	NULLABLE
NAME	Y
BREED	N

```
> insert into pedigree_pooches (name, breed) values ('Franck', null);
```

SQL Error: ORA-01400: cannot insert NULL into ("PDBADMIN"."PEDIGREE_POOCHES"."BREED")
*Cause: An attempt was made to insert NULL into previously listed objects.
*Action: These objects cannot accept NULL values. Reservable columns cannot accept NULL values.

```
> create table pedigree_pooches(name varchar2(16),
2*                      breed varchar2(16) not null);
```

Table PEDIGREE_POOCHES created.

```
> select constraint_name, search_condition from user_constraints where table_name = 'PEDIGREE_POOCHES';
```

CONSTRAINT_NAME	SEARCH_CONDITION
SYS_C008494	"BREED" IS NOT NULL

```
> select column_name, nullable from user_tab_cols where table_name = 'PEDIGREE_POOCHES';
```

COLUMN_NAME	NULLABLE
NAME	Y
BREED	N

```
> insert into pedigree_pooches (name, breed) values ('Franck', null);
```

SQL Error: ORA-01400: cannot insert NULL into ("PDBADMIN"."PEDIGREE_POOCHES"."BREED")
*Cause: An attempt was made to insert NULL into previously listed objects.
*Action: These objects cannot accept NULL values. Reservable columns cannot accept NULL values.

```
> select count(name) from pedigree_pooches where breed = 'Spaniel';
```

COUNT(NAME)

6

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1	1	1	14
1	SORT AGGREGATE		1	1	1	14
2	TABLE ACCESS BY INDEX ROWID BATCHED	PEDIGREE_POOCHES	1	6	6	14
* 3	INDEX RANGE SCAN	POOCH_BREEDS	1	6	6	2

```
> select count(*) from pedigree_pooches where name is null;
```

COUNT(*)

0

```
> select count(name) from pedigree_pooches where breed = 'Spaniel';
```

COUNT(NAME)

6

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1	1	1	14
1	SORT AGGREGATE		1	1	1	14
2	TABLE ACCESS BY INDEX ROWID BATCHED	PEDIGREE_POOCHES	1	6	6	14
* 3	INDEX RANGE SCAN	POOCH_BREEDS	1	6	6	2

```
> select count(*) from pedigree_pooches where name is null;
```

COUNT(*)

0

```
> select count(name) from pedigree_pooches where breed = 'Spaniel';
```

COUNT(NAME)

6

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1	1	1	14
1	SORT AGGREGATE		1	1	1	14
2	TABLE ACCESS BY INDEX ROWID BATCHED	PEDIGREE_POOCHES	1	6	6	14
* 3	INDEX RANGE SCAN	POOCH_BREEDS	1	6	6	2

```
> select count(*) from pedigree_pooches where name is null;
```

COUNT(*)

0

```
> select count(name) from pedigree_pooches where breed = 'Spaniel';
```

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1		1	14
1	SORT AGGREGATE		1	1	1	14
2	TABLE ACCESS BY INDEX ROWID BATCHED	PEDIGREE_POOCHES	1	6	6	14
* 3	INDEX RANGE SCAN	POOCH_BREEDS	1	6	6	2

```
> alter table pedigree_pooches modify name not null;
```

Table PEDIGREE_POOCHES altered.

```
> select count(name) from pedigree_pooches where breed = 'Spaniel';
```

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1		1	2
1	SORT AGGREGATE		1	1	1	2
* 2	INDEX RANGE SCAN	POOCH_BREEDS	1	6	6	2

```
> select count(name) from pedigree_pooches where breed = 'Spaniel';
```

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1		1	14
1	SORT AGGREGATE		1	1	1	14
2	TABLE ACCESS BY INDEX ROWID BATCHED	PEDIGREE_POOCHES	1	6	6	14
* 3	INDEX RANGE SCAN	POOCH_BREEDS	1	6	6	2

```
> alter table pedigree_pooches modify name not null;
```

Table PEDIGREE_POOCHES altered.

```
> select count(name) from pedigree_pooches where breed = 'Spaniel';
```

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1		1	2
1	SORT AGGREGATE		1	1	1	2
* 2	INDEX RANGE SCAN	POOCH_BREEDS	1	6	6	2

```
> select count(name) from pedigree_pooches where breed = 'Spaniel';
```

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1		1	14
1	SORT AGGREGATE		1	1	1	14
2	TABLE ACCESS BY INDEX ROWID BATCHED	PEDIGREE_POOCHES	1	6	6	14
* 3	INDEX RANGE SCAN	POOCH_BREEDS	1	6	6	2

```
> alter table pedigree_pooches modify name not null;
```

Table PEDIGREE_POOCHES altered.

```
> select count(name) from pedigree_pooches where breed = 'Spaniel';
```

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1		1	2
1	SORT AGGREGATE		1	1	1	2
* 2	INDEX RANGE SCAN	POOCH_BREEDS	1	6	6	2

```
> create table bad_dogs(name varchar2(16) constraint name_nn not null,  
2*                      owner varchar2(16) check (owner is not null) );
```

Table BAD_DOGS created.

```
> create table people(name varchar(16) constraint name_nn not null);
```

ORA-02264: name already used by an existing constraint

```
SQL> insert into bad_dogs(name, owner) values ('Gnasher', null);
```

ORA-02290: check constraint (PDBADMIN.SYS_C008491) violated

```
> select nullable from user_tab_cols where table_name = 'BAD_DOGS' and column_name = 'OWNER';
```

NULLABLE

Y

```
> create table good_dogs(name varchar2(16) not null);
```

Table GOOD_DOGS created.

```
> create table bad_dogs(name varchar2(16) constraint name_nn not null,  
2*                      owner varchar2(16) check (owner is not null) );
```

Table BAD_DOGS created.

```
> create table people(name varchar(16) constraint name_nn not null);
```

ORA-02264: name already used by an existing constraint

```
SQL> insert into bad_dogs(name, owner) values ('Gnasher', null);
```

ORA-02290: check constraint (PDBADMIN.SYS_C008491) violated

```
> select nullable from user_tab_cols where table_name = 'BAD_DOGS' and column_name = 'OWNER';
```

NULLABLE

Y

```
> create table good_dogs(name varchar2(16) not null);
```

Table GOOD_DOGS created.

```
> create table bad_dogs(name varchar2(16) constraint name_nn not null,  
2*                      owner varchar2(16) check (owner is not null) );
```

Table BAD_DOGS created.

```
> create table people(name varchar(16) constraint name_nn not null);
```

ORA-02264: name already used by an existing constraint

```
SQL> insert into bad_dogs(name, owner) values ('Gnasher', null);
```

ORA-02290: check constraint (PDBADMIN.SYS_C008491) violated

```
> select nullable from user_tab_cols where table_name = 'BAD_DOGS' and column_name = 'OWNER';
```

NULLABLE

Y

```
> create table good_dogs(name varchar2(16) not null);
```

Table GOOD_DOGS created.

```
> create table bad_dogs(name varchar2(16) constraint name_nn not null,  
2*                      owner varchar2(16) check (owner is not null) );
```

Table BAD_DOGS created.

```
> create table people(name varchar(16) constraint name_nn not null);
```

ORA-02264: name already used by an existing constraint

```
SQL> insert into bad_dogs(name, owner) values ('Gnasher', null);
```

ORA-02290: check constraint (PDBADMIN.SYS_C008491) violated

```
> select nullable from user_tab_cols where table_name = 'BAD_DOGS' and column_name = 'OWNER';
```

NULLABLE

Y

```
> create table good_dogs(name varchar2(16) not null);
```

Table GOOD_DOGS created.

```
> create table bad_dogs(name varchar2(16) constraint name_nn not null,  
2*                      owner varchar2(16) check (owner is not null) );
```

Table BAD_DOGS created.

```
> create table people(name varchar(16) constraint name_nn not null);
```

ORA-02264: name already used by an existing constraint

```
SQL> insert into bad_dogs(name, owner) values ('Gnasher', null);
```

ORA-02290: check constraint (PDBADMIN.SYS_C008491) violated

```
> select nullable from user_tab_cols where table_name = 'BAD_DOGS' and column_name = 'OWNER';
```

NULLABLE

Y

```
> create table good_dogs(name varchar2(16) not null);
```

Table GOOD_DOGS created.



Primary Keys

```
> create table dogs(name varchar2(15) primary key,  
2*           owner varchar2(15) null      );
```

Table DOGS created.

```
> select nullable from user_tab_cols where table_name = 'DOGS' and column_name = 'NAME';
```

NULLABLE

N

```
> select constraint_name, constraint_type from user_constraints where table_name = 'DOGS';
```

CONSTRAINT_NAME	CONSTRAINT_TYPE
SYS_C008513	P

```
> create table dogs(name varchar2(15) primary key,  
2*           owner varchar2(15) null      );
```

Table DOGS created.

```
> select nullable from user_tab_cols where table_name = 'DOGS' and column_name = 'NAME';
```

NULLABLE

```
-----  
N
```

```
> select constraint_name, constraint_type from user_constraints where table_name = 'DOGS';
```

CONSTRAINT_NAME	CONSTRAINT_TYPE
SYS_C008513	P

```
> create table dogs(name varchar2(15) primary key,  
2*           owner varchar2(15) null      );
```

Table DOGS created.

```
> select nullable from user_tab_cols where table_name = 'DOGS' and column_name = 'NAME';
```

NULLABLE

```
-----  
N
```

```
> select constraint_name, constraint_type from user_constraints where table_name = 'DOGS';
```

CONSTRAINT_NAME	CONSTRAINT_TYPE
SYS_C008513	P

```
> alter table dogs disable primary key;
```

Table DOGS altered.

```
> select nullable from user_tab_cols where table_name = 'DOGS' and column_name = 'NAME';
```

NULLABLE

```
-----
```

Y

```
SQL> alter table dogs modify name not null enable enable primary key;
```

Table DOGS altered.

```
> alter table dogs disable primary key;
```

Table DOGS altered.

```
> select nullable from user_tab_cols where table_name = 'DOGS' and column_name = 'NAME';
```

NULLABLE

```
-----  
Y
```

```
SQL> alter table dogs modify name not null enable enable primary key;
```

Table DOGS altered.

```
> alter table dogs disable primary key;
```

Table DOGS altered.

```
> select nullable from user_tab_cols where table_name = 'DOGS' and column_name = 'NAME';
```

NULLABLE

Y

```
SQL> alter table dogs modify name not null enable enable primary key;
```

Table DOGS altered.



Foreign Keys

```
> create table people(name varchar2(16) not null primary key);
```

Table PEOPLE created.

```
> insert into people (name) values ('Martin'),  
      ('Martin');
```

ORA-00001: unique constraint (PDBADMIN.SYS_C008427) violated on table PDBADMIN.PEOPLE columns (NAME)

ORA-03301: (ORA-00001 details) row with column values (NAME:'Martin') already exists

```
> alter table dogs add constraint fk_dog_owners foreign key (owner) references people(name);
```

Table DOGS altered.

```
> create table people(name varchar2(16) not null primary key);
```

Table PEOPLE created.

```
> insert into people (name) values ('Martin'),  
      ('Martin');
```

ORA-00001: unique constraint (PDBADMIN.SYS_C008427) violated on table PDBADMIN.PEOPLE columns (NAME)

ORA-03301: (ORA-00001 details) row with column values (NAME:'Martin') already exists

```
> alter table dogs add constraint fk_dog_owners foreign key (owner) references people(name);
```

Table DOGS altered.

```
> create table people(name varchar2(16) not null primary key);
```

Table PEOPLE created.

```
> insert into people (name) values ('Martin'),  
      ('Martin');
```

ORA-00001: unique constraint (PDBADMIN.SYS_C008427) violated on table PDBADMIN.PEOPLE columns (NAME)

ORA-03301: (ORA-00001 details) row with column values (NAME:'Martin') already exists

```
> alter table dogs add constraint fk_dog_owners foreign key (owner) references people(name);
```

Table DOGS altered.



```
> insert into dogs (name, owner )  
2      values ('Nala', 'Manuela');
```

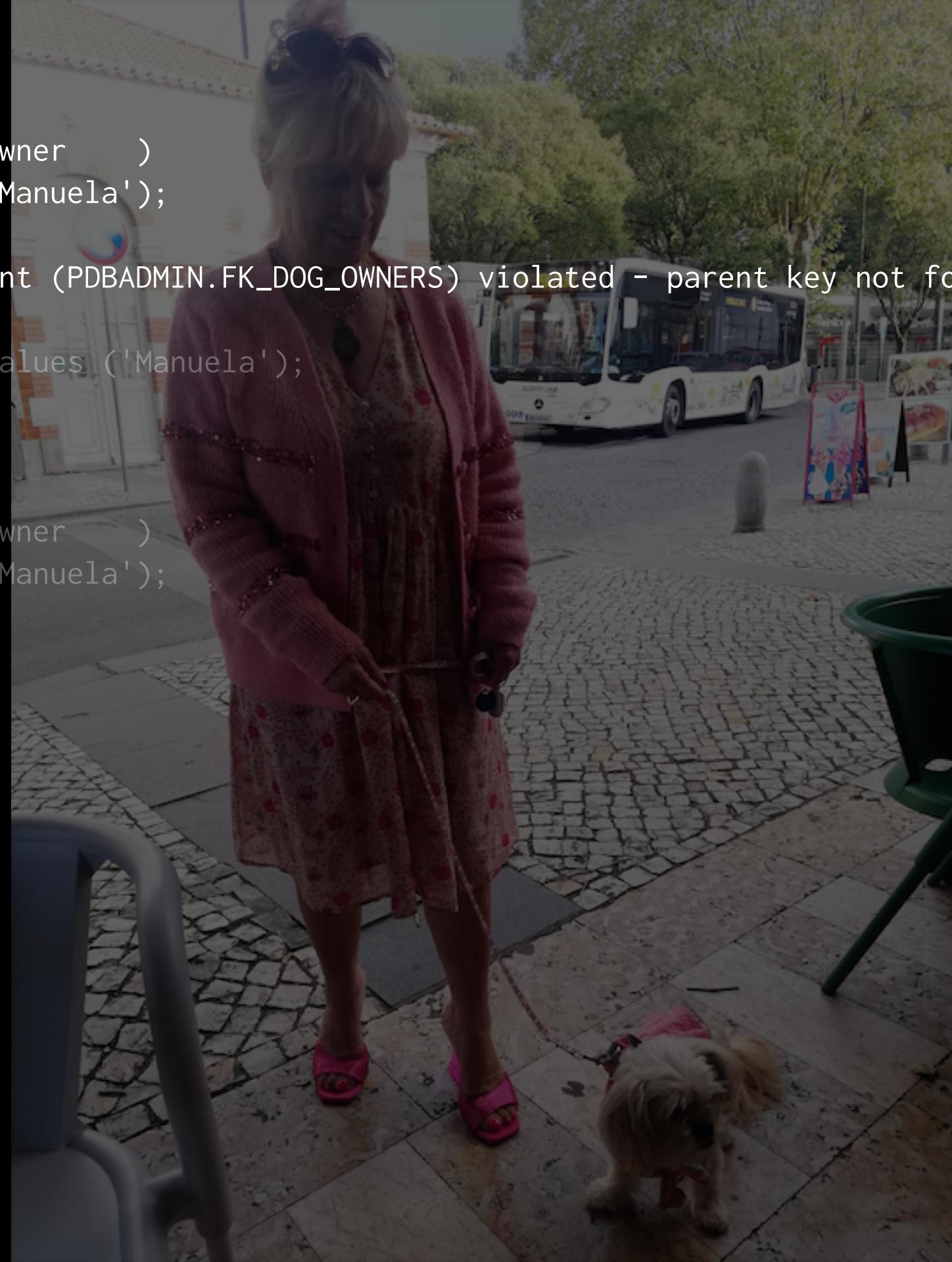
ORA-02291: integrity constraint (PDBADMIN.FK_DOG OWNERS) violated - parent key not found

```
> insert into people (name) values ('Manuela');
```

1 row inserted.

```
> insert into dogs (name, owner )  
2      values ('Nala', 'Manuela');
```

1 row inserted.



```
> insert into dogs (name, owner )  
2      values ('Nala', 'Manuela');
```

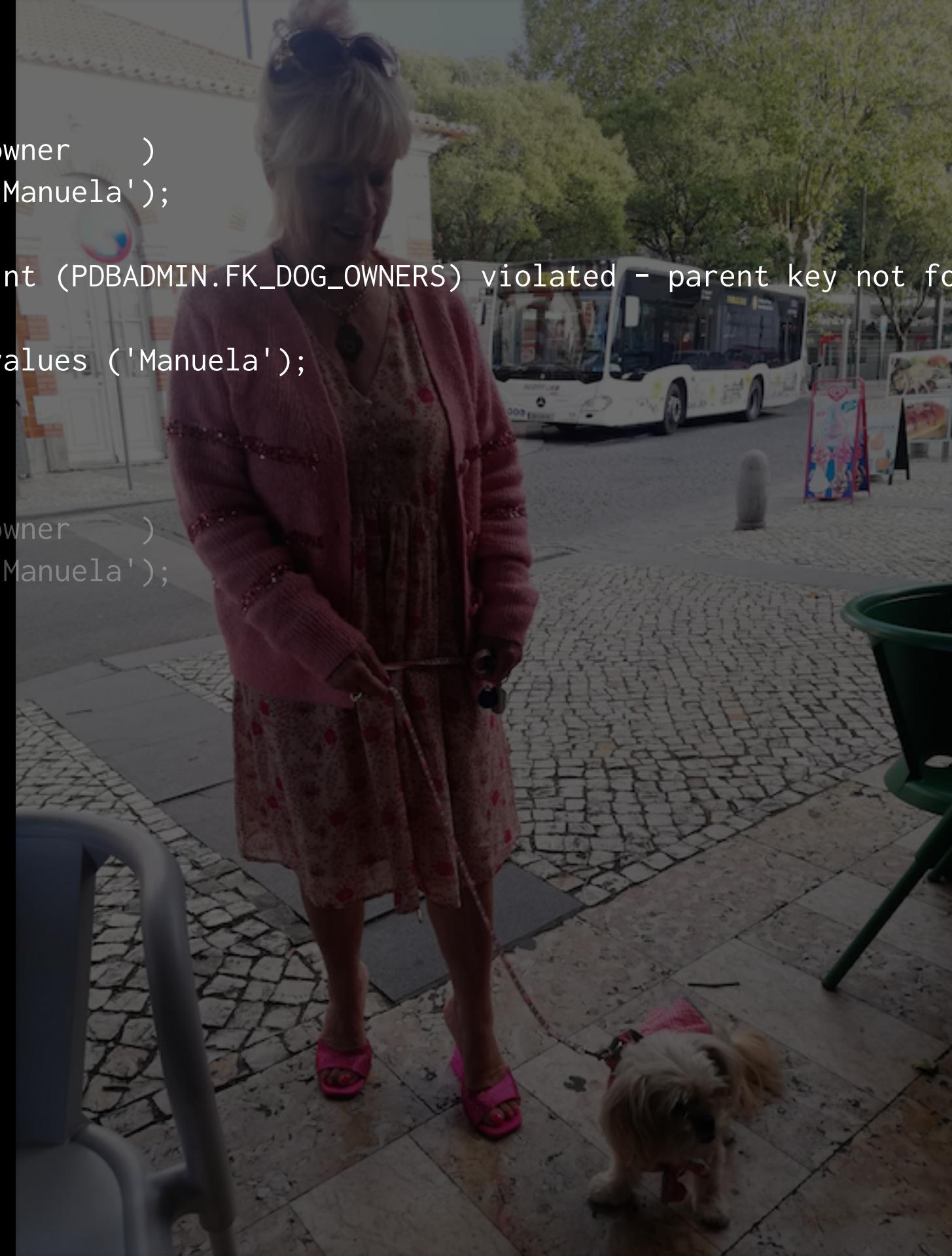
ORA-02291: integrity constraint (PDBADMIN.FK_DOG OWNERS) violated - parent key not found

```
> insert into people (name) values ('Manuela');
```

1 row inserted.

```
> insert into dogs (name, owner )  
2      values ('Nala', 'Manuela');
```

1 row inserted.



```
> insert into dogs (name, owner )  
2      values ('Nala', 'Manuela');
```

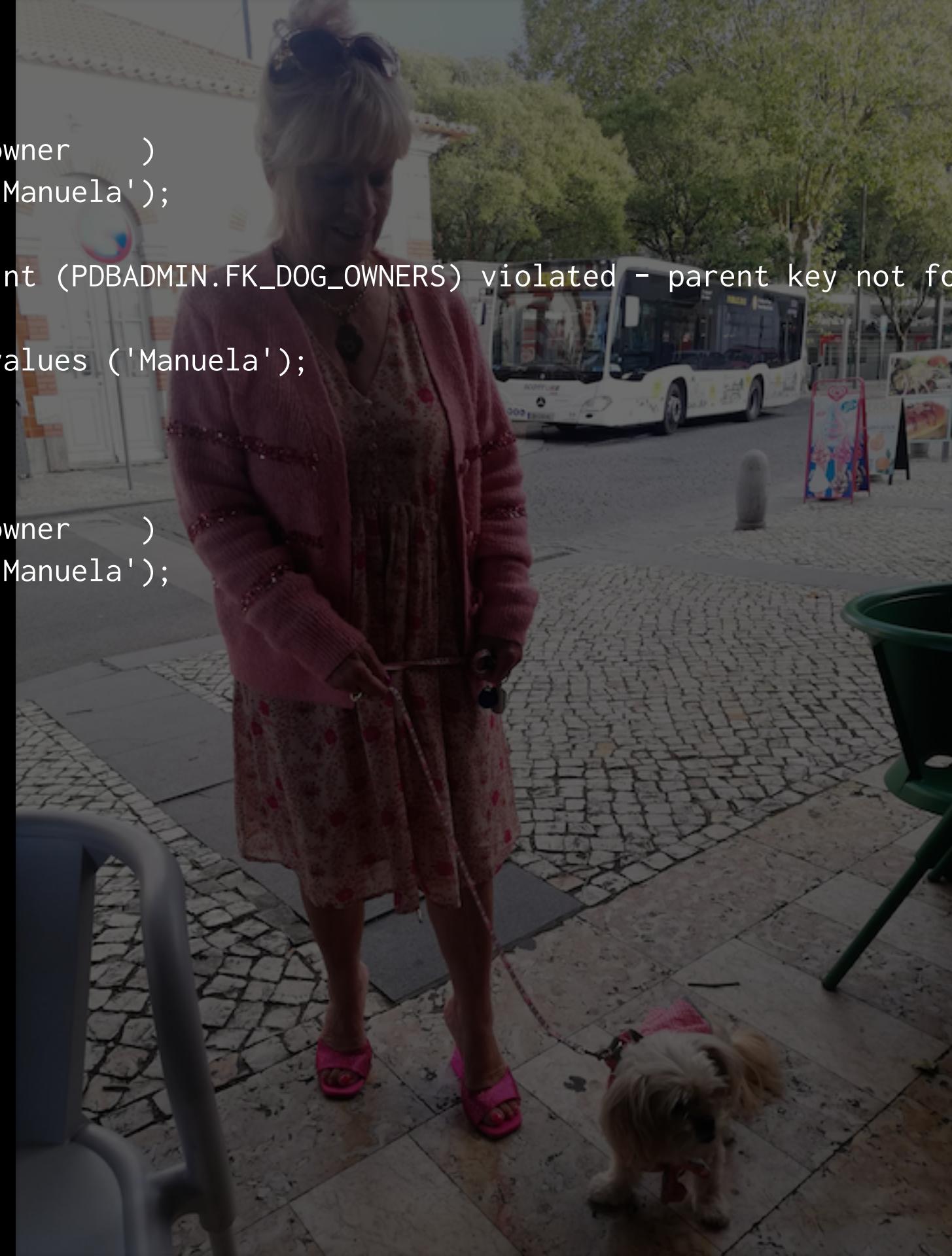
ORA-02291: integrity constraint (PDBADMIN.FK_DOG OWNERS) violated - parent key not found

```
> insert into people (name) values ('Manuela');
```

1 row inserted.

```
> insert into dogs (name, owner )  
2      values ('Nala', 'Manuela');
```

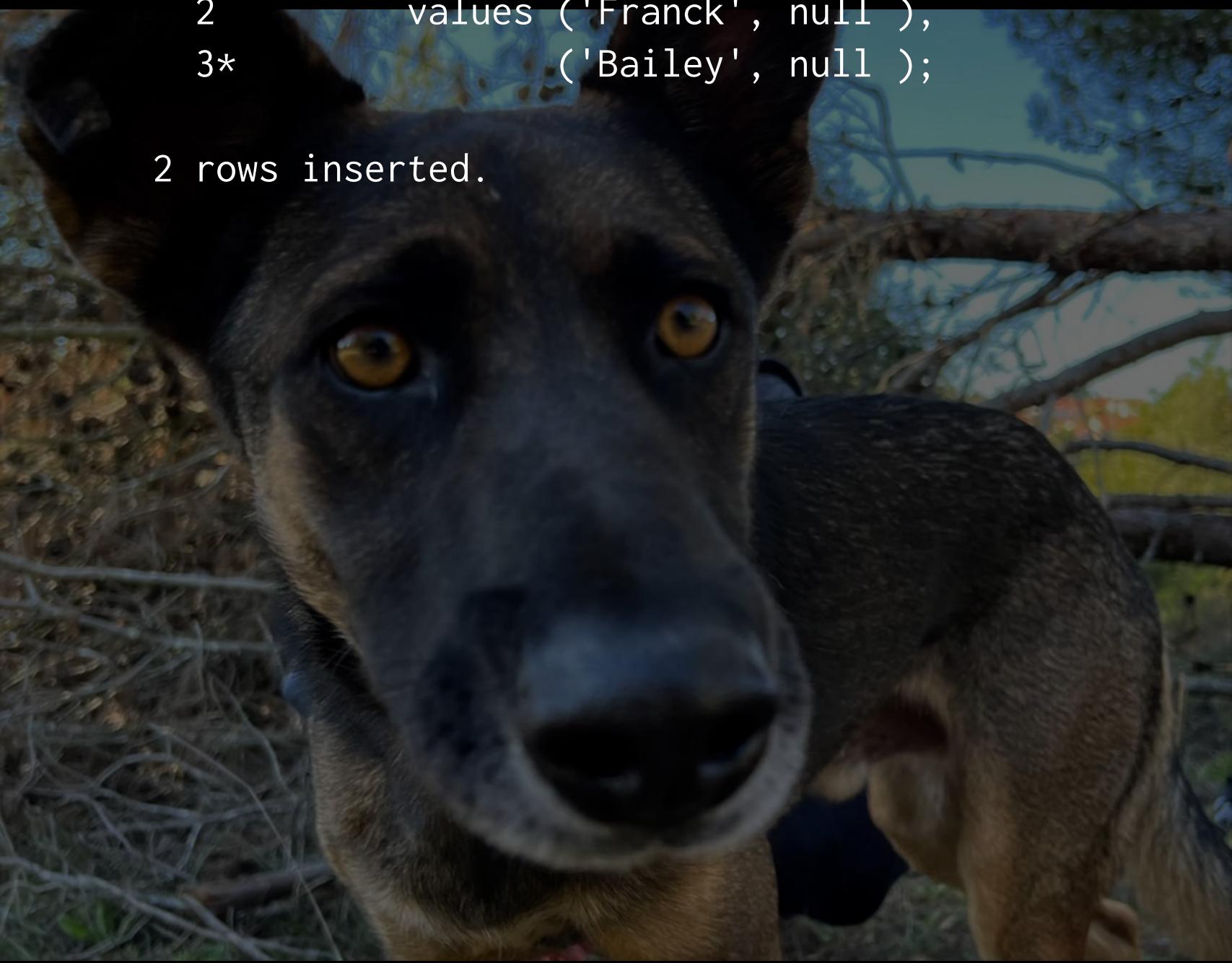
1 row inserted.





```
> insert into dogs (name, owner)  
2      values ('Franck', null ),  
3*      ('Bailey', null );
```

2 rows inserted.



> select name from people;

NAME

Manuela

> select name, owner from dogs;

NAME

OWNER

Nala

Manuela

Franck

[null]

Bailey

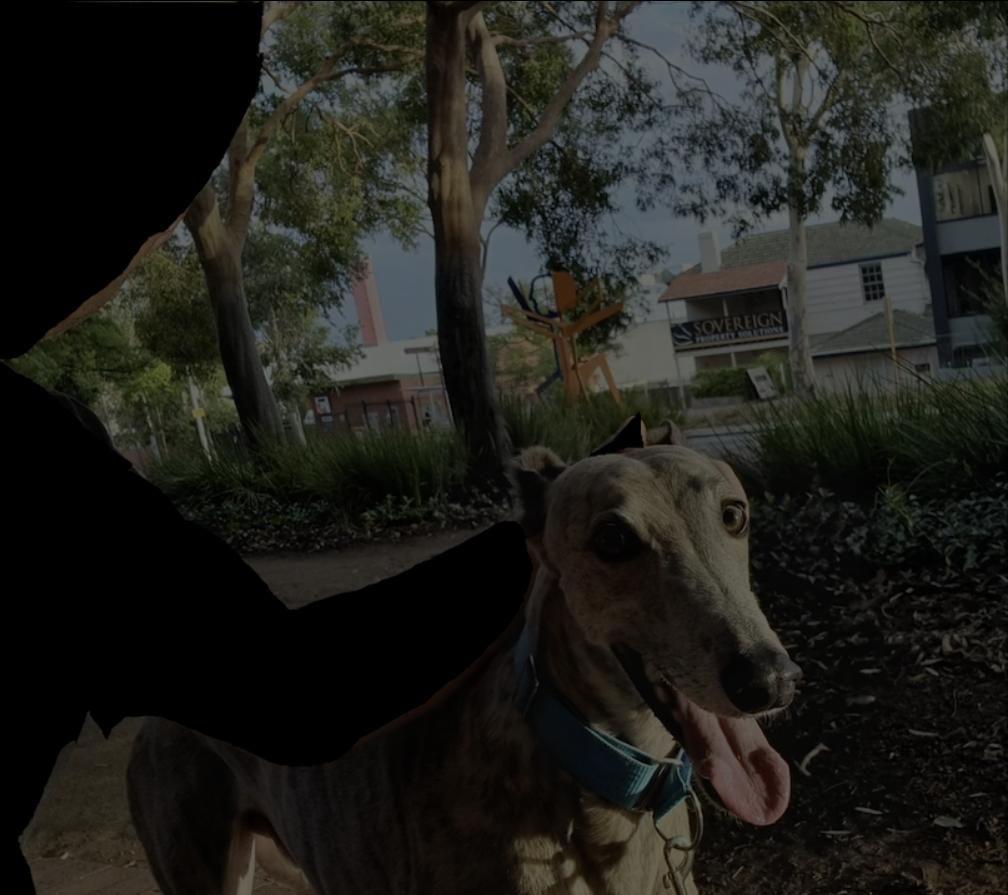
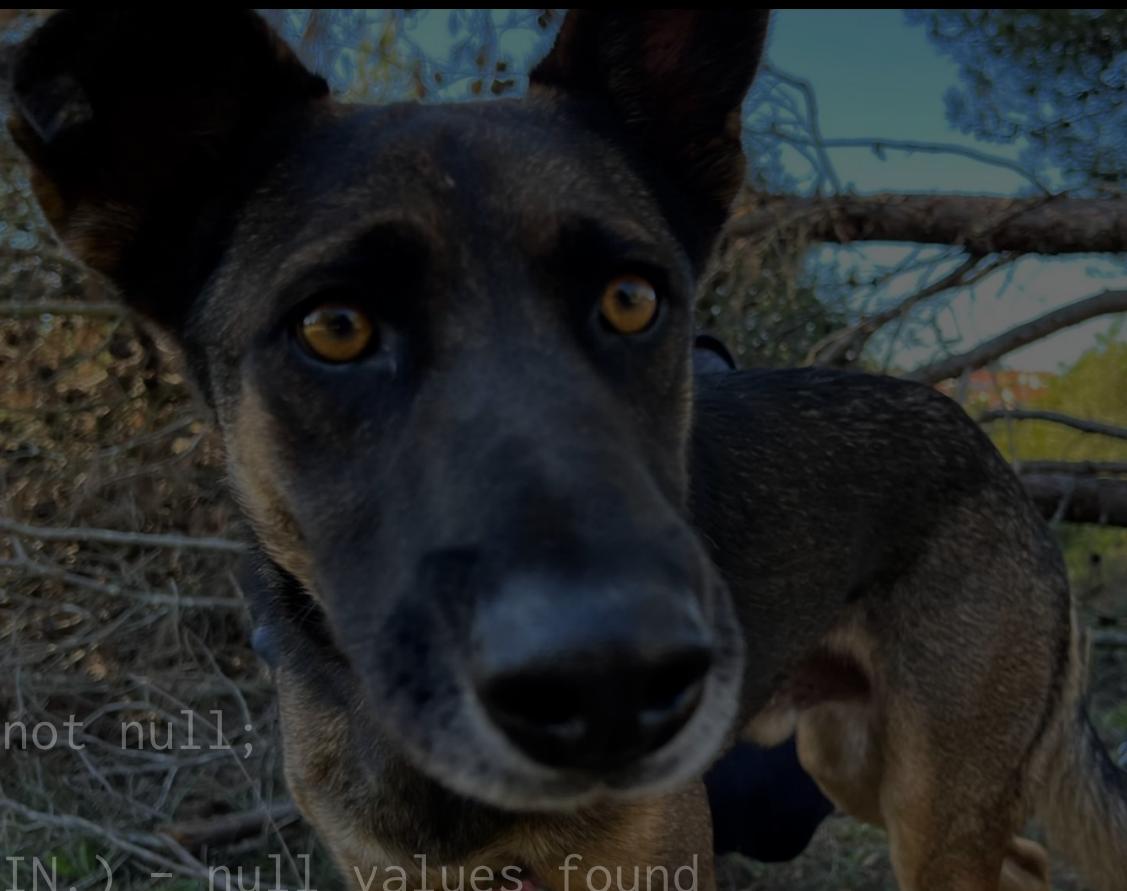
[null]

> alter table dogs modify owner not null;

ORA-02296: cannot enable (PDBADMIN.) - null values found

*Cause: an alter table enable constraint failed because the table contains values that do not satisfy the constraint.

*Action: Obvious



```
> select name from people;
```

NAME

Manuela

```
> select name, owner from dogs;
```

NAME

OWNER

Nala

Manuela

Franck

[null]

Bailey

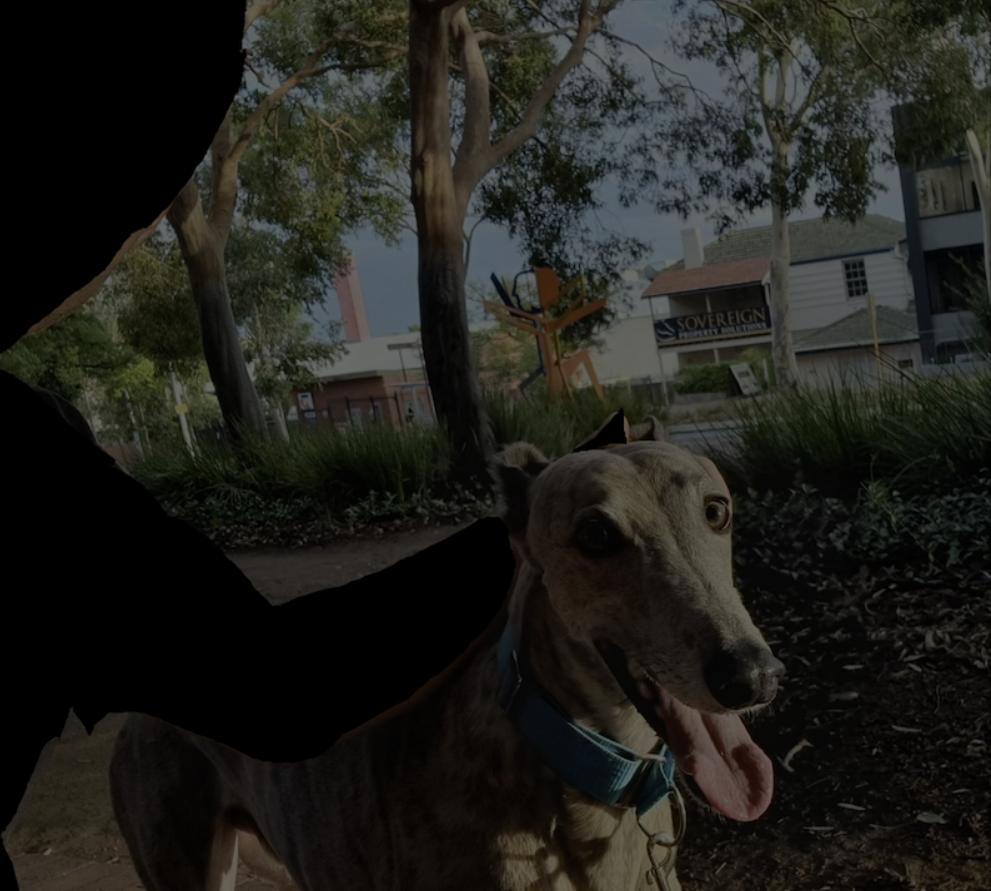
[null]

```
> alter table dogs modify owner not null;
```

ORA-02296: cannot enable (PDBADMIN.) - null values found

*Cause: an alter table enable constraint failed because the table contains values that do not satisfy the constraint.

*Action: Obvious



> select name from people;

NAME

Manuela

> select name, owner from dogs;

NAME

OWNER

Nala

Manuela

Franck

[null]

Bailey

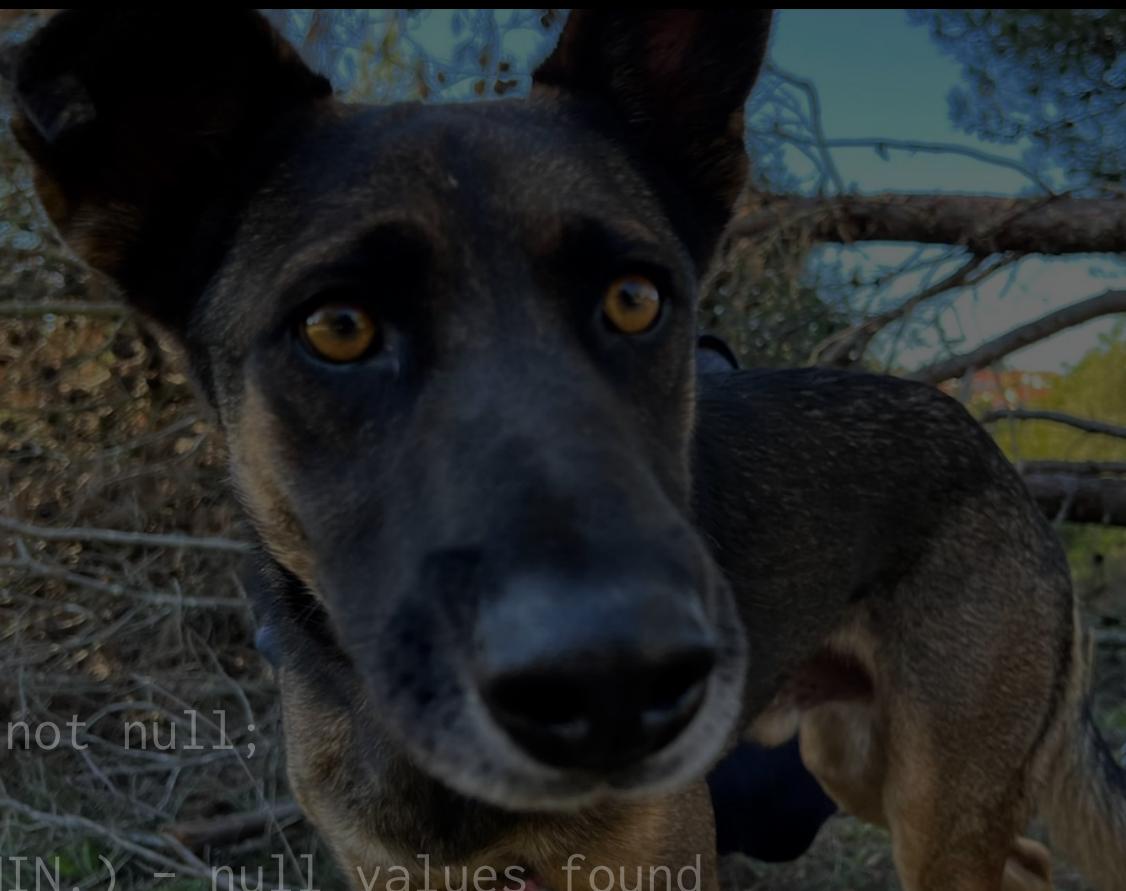
[null]

> alter table dogs modify owner not null;

ORA-02296: cannot enable (PDBADMIN.) - null values found

*Cause: an alter table enable constraint failed because the table contains values that do not satisfy the constraint.

*Action: Obvious

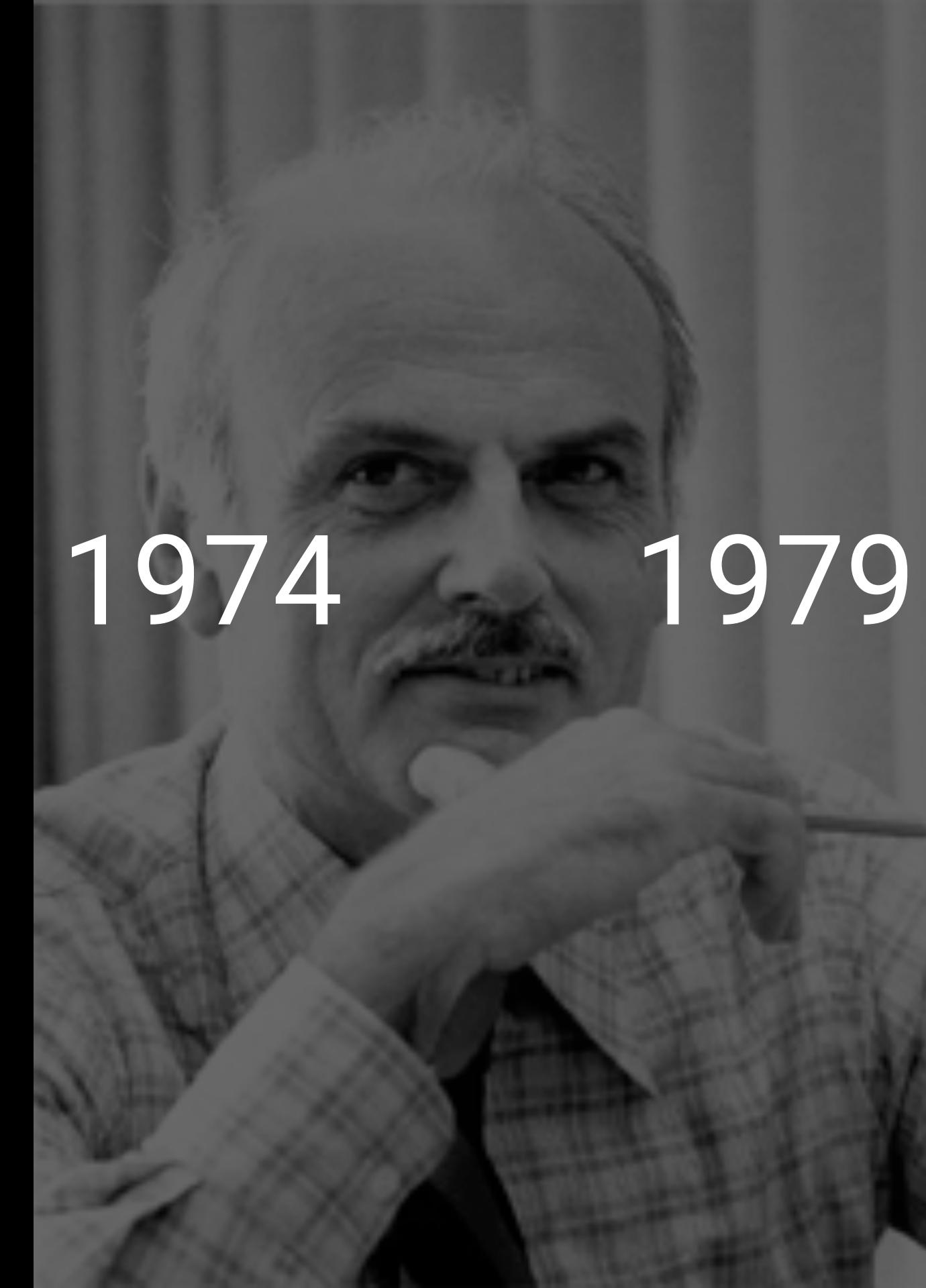


1970

1974

1979

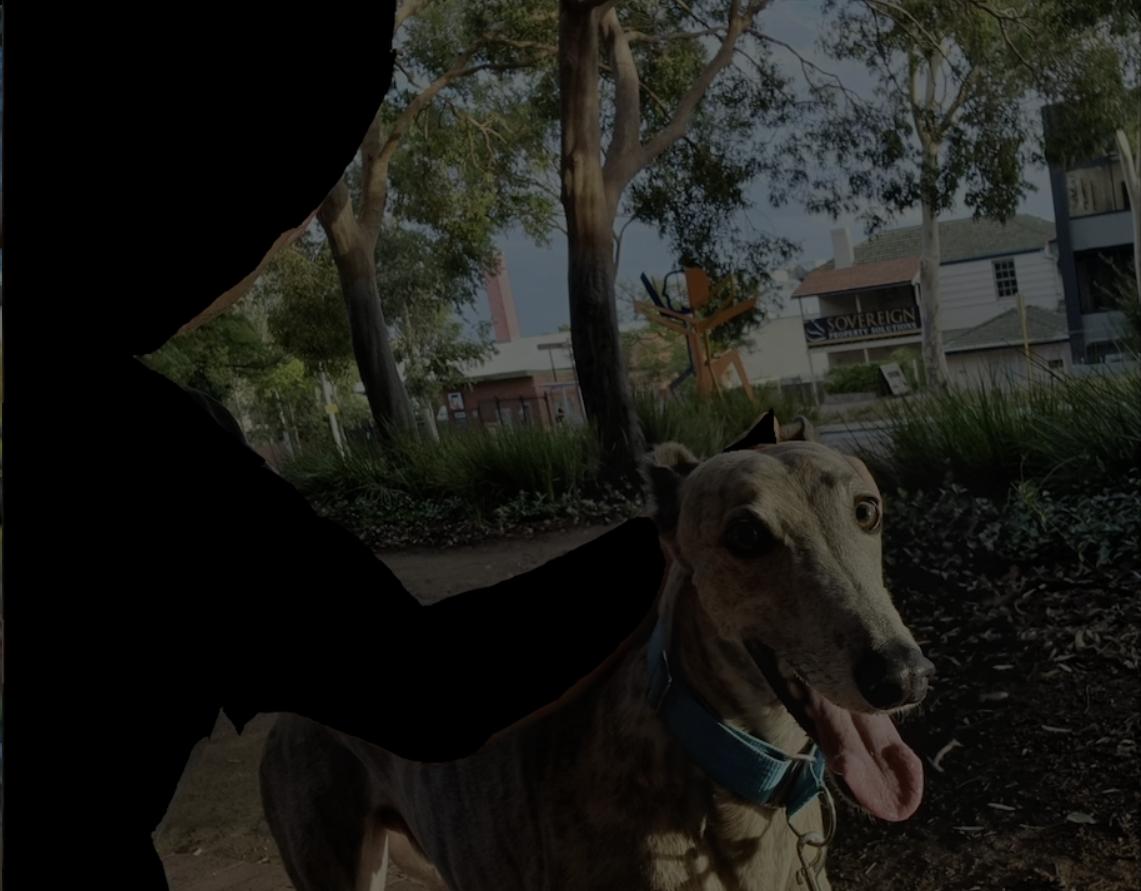
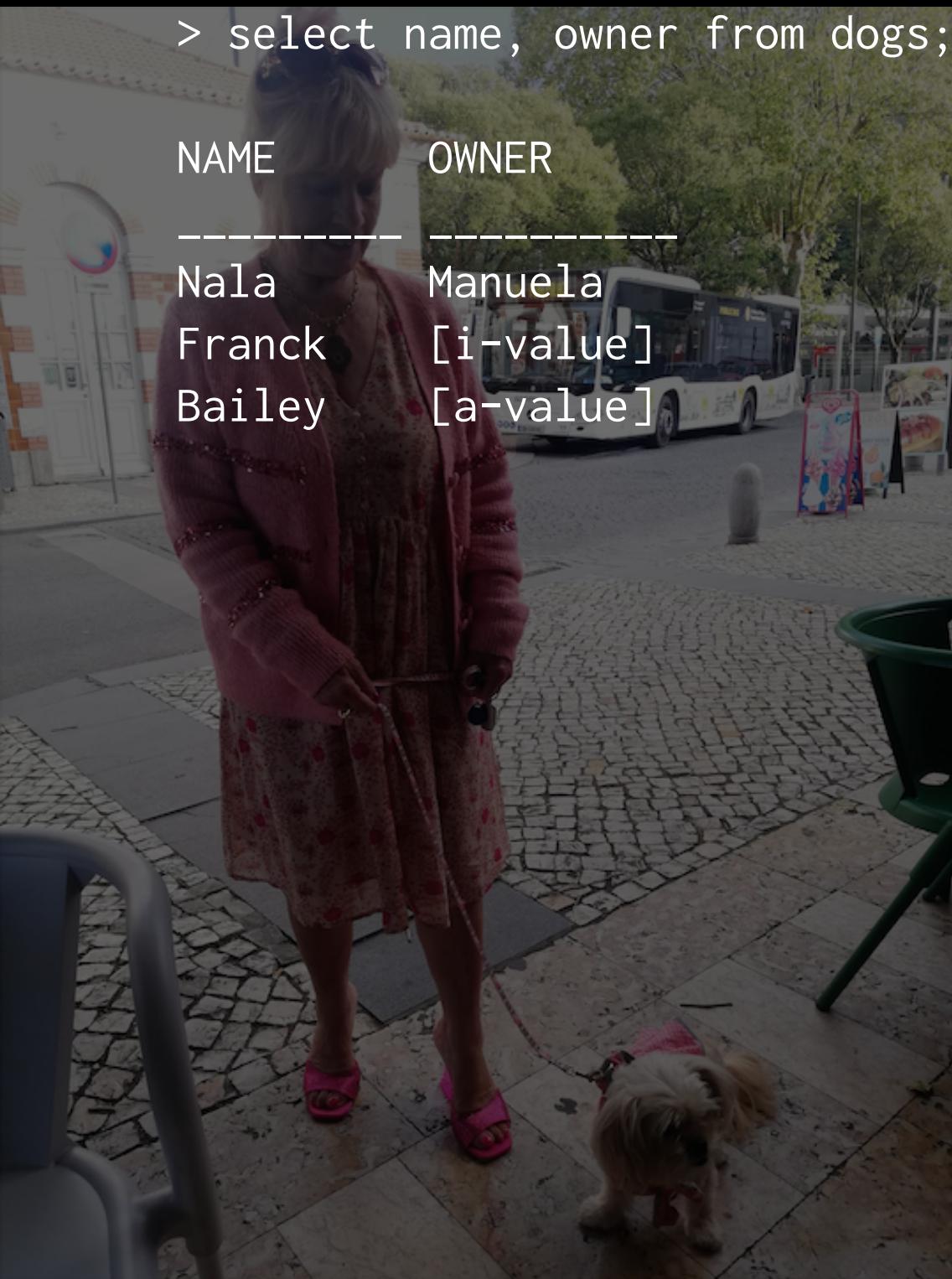
1986



> select name, owner from dogs;

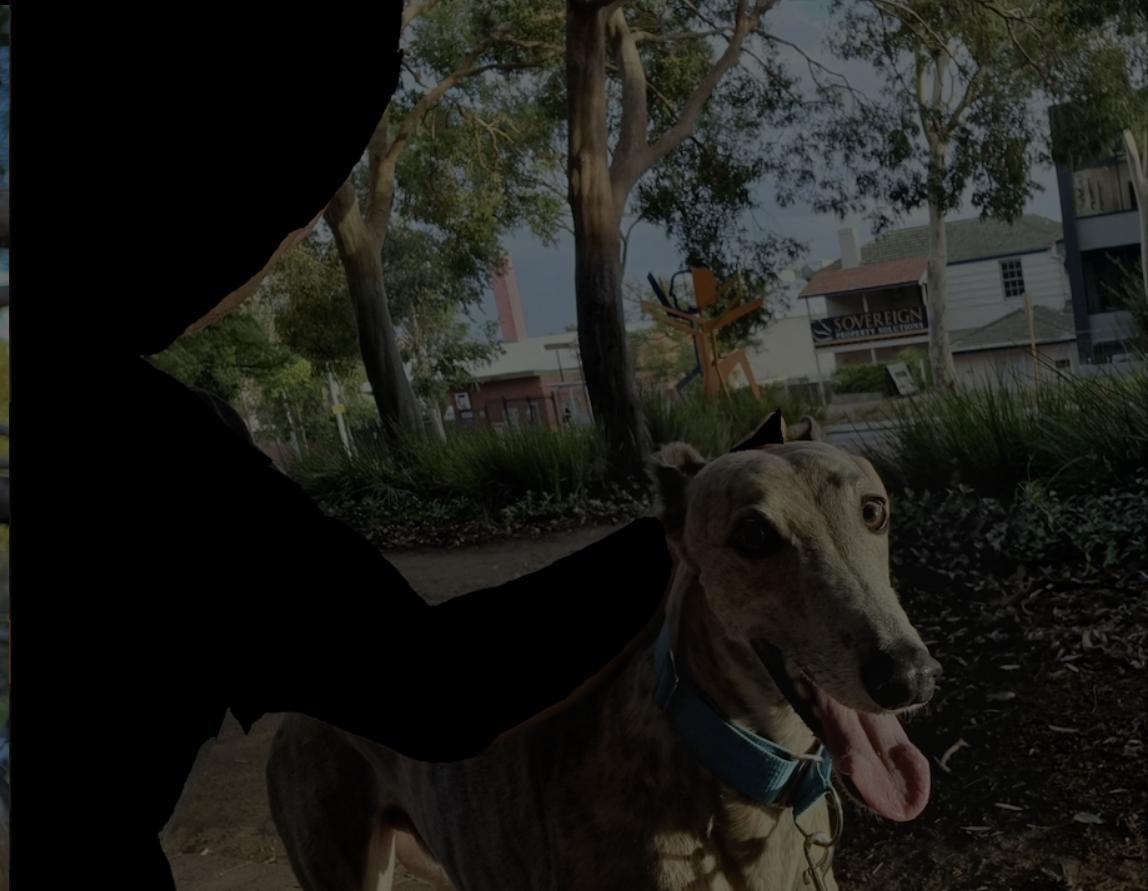
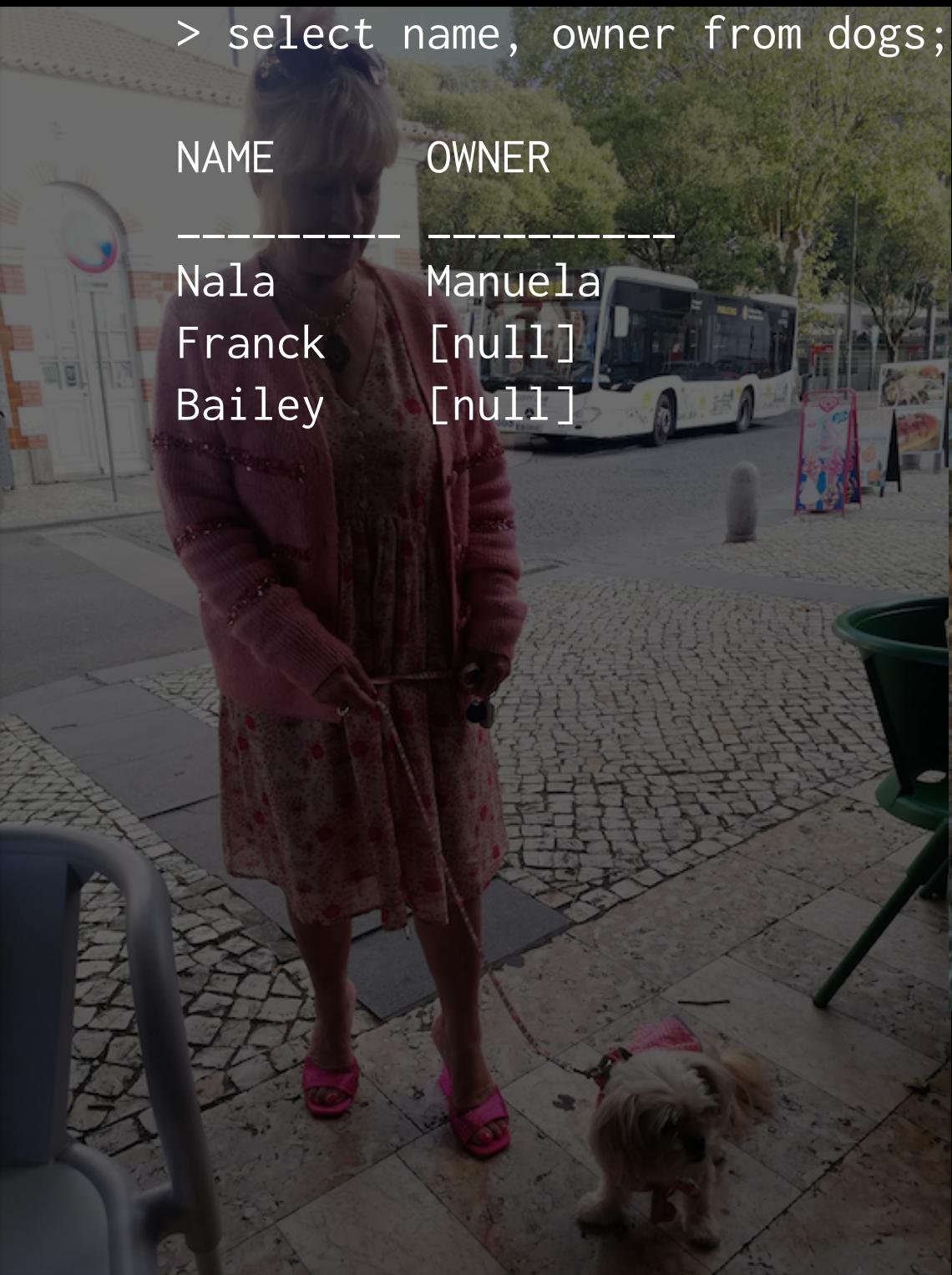
NAME OWNER

Nala Manuela
Franck [i-value]
Bailey [a-value]



> select name, owner from dogs;

NAME	OWNER
Nala	Manuela
Franck	[null]
Bailey	[null]



Unique Constraints

```
> alter table dogs add handle varchar2(16);
```

Table DOGS altered.

```
> alter table dogs add constraint unique_handle unique(handle);
```

Table DOGS altered.

```
> alter table dogs add handle varchar2(16);
```

Table DOGS altered.

```
> alter table dogs add constraint unique_handle unique(handle);
```

Table DOGS altered.



@dougtwitter / doug the Pug

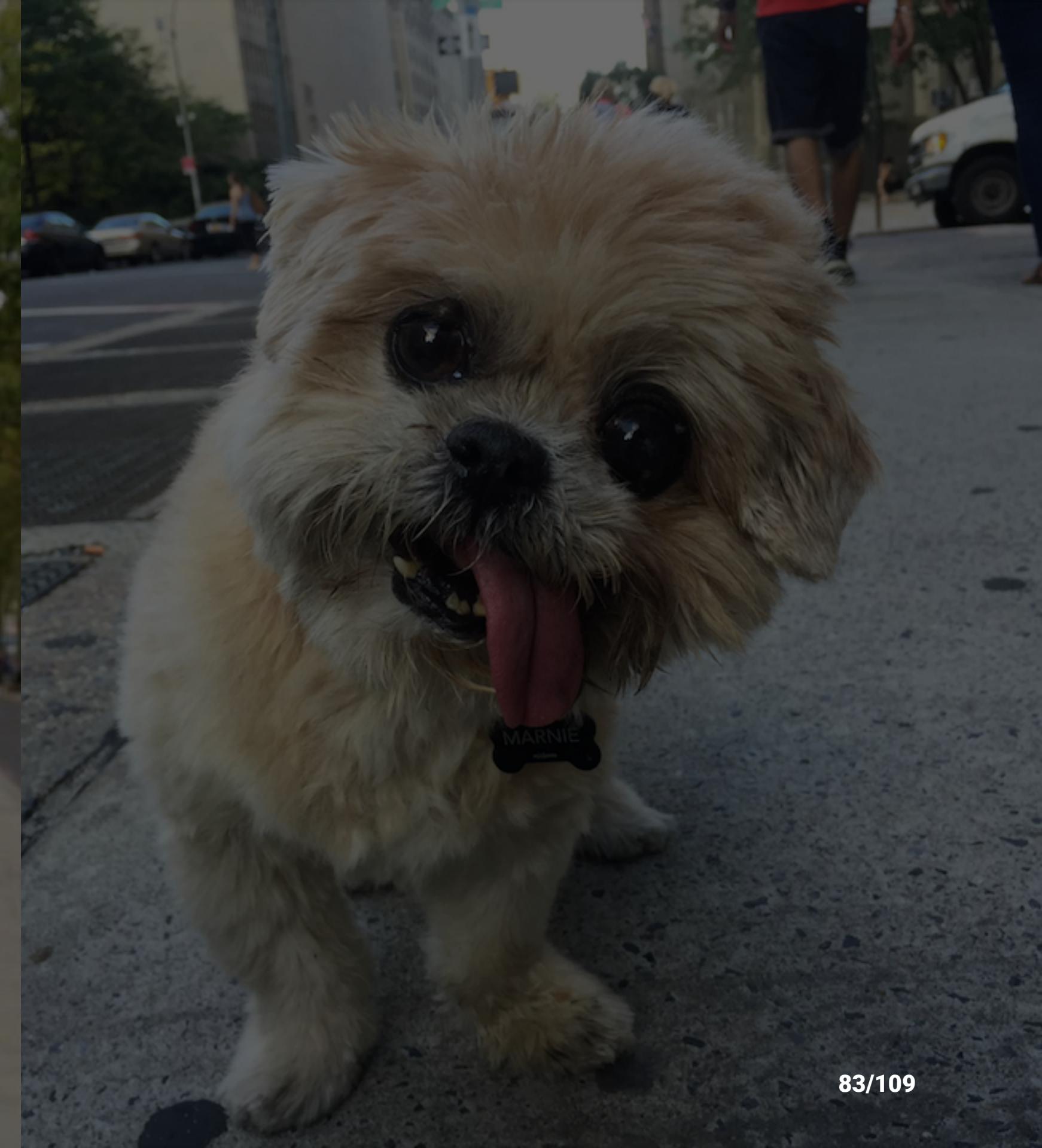
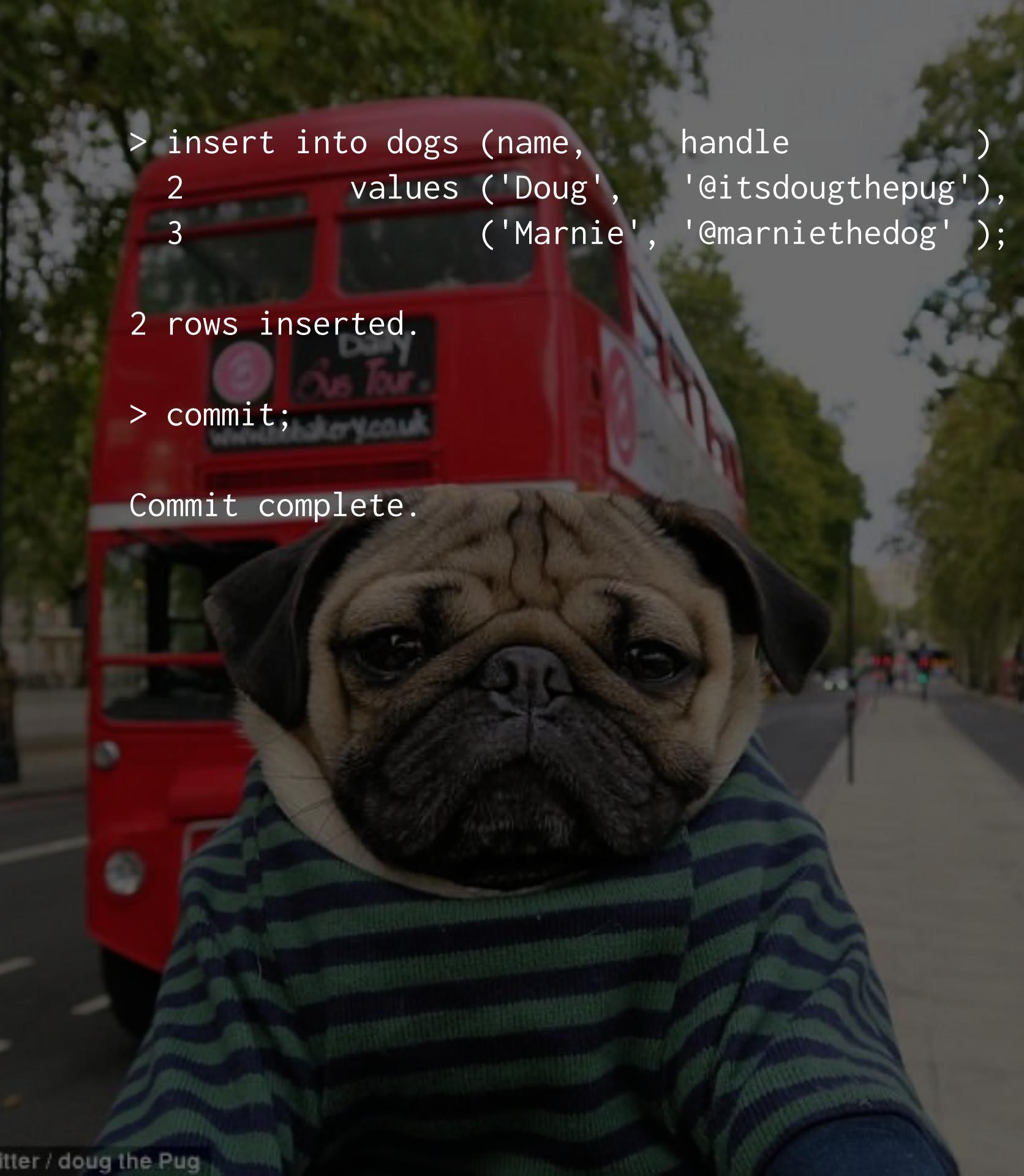


```
> insert into dogs (name, handle  
2      values ('Doug', '@itsdougthepug'),  
3      ('Marnie', '@marniethedog' );
```

2 rows inserted.

```
> commit;
```

Commit complete.

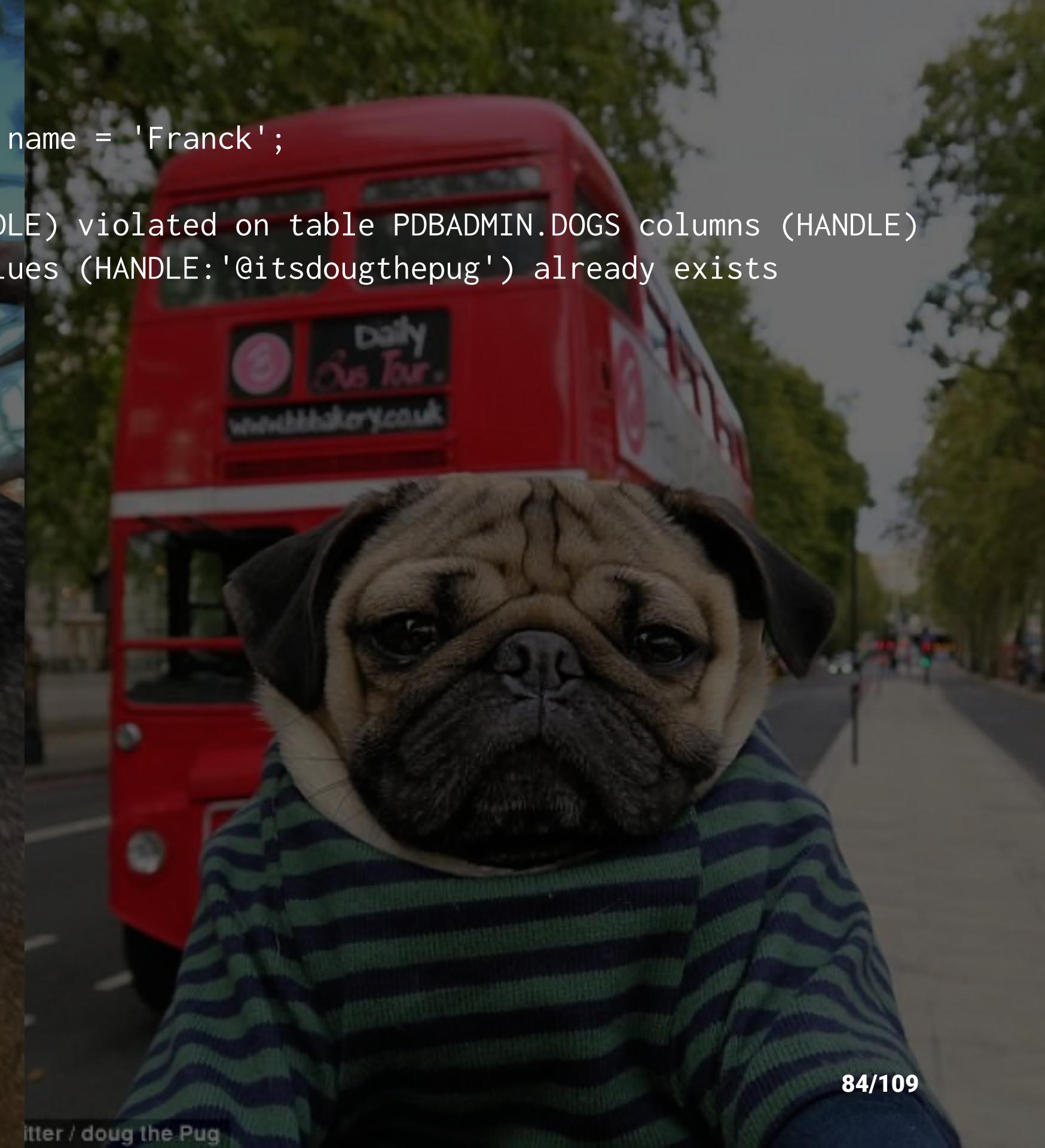


```
> update dogs set handle = '@itsdougthezug' where name = 'Franck';
```

```
ORA-00001: unique constraint (PDBADMIN.UNIQUE_HANDLE) violated on table PDBADMIN.DOGS columns (HANDLE)  
ORA-03301: (ORA-00001 details) row with column values (HANDLE:'@itsdougthezug') already exists
```

```
> select name, handle from dogs order by handle;
```

NAME	HANDLE
Doug	@itsdougthezug
Marnie	@marniethedog
Bailey	[null]
Nala	[null]
Franck	[null]

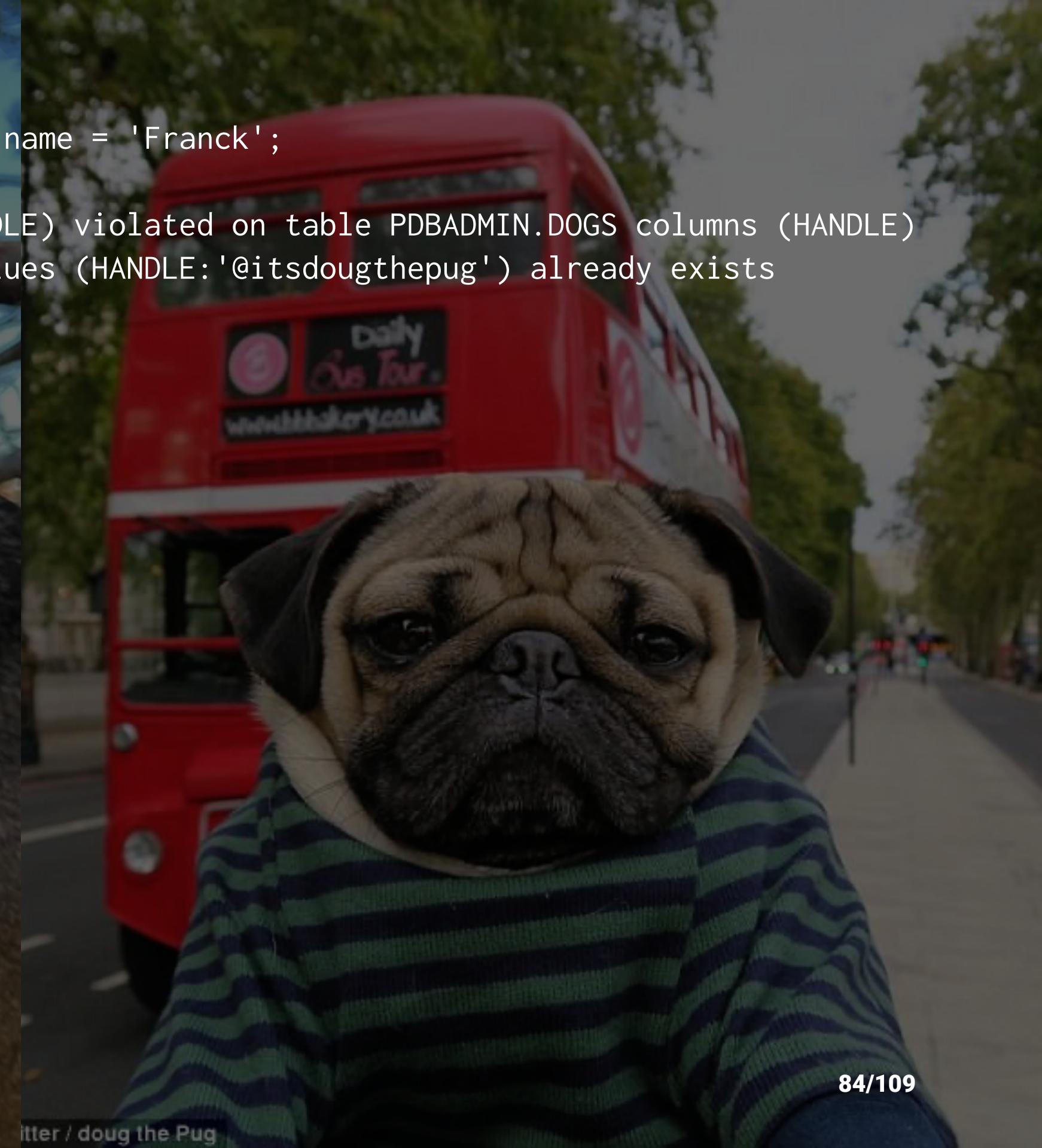


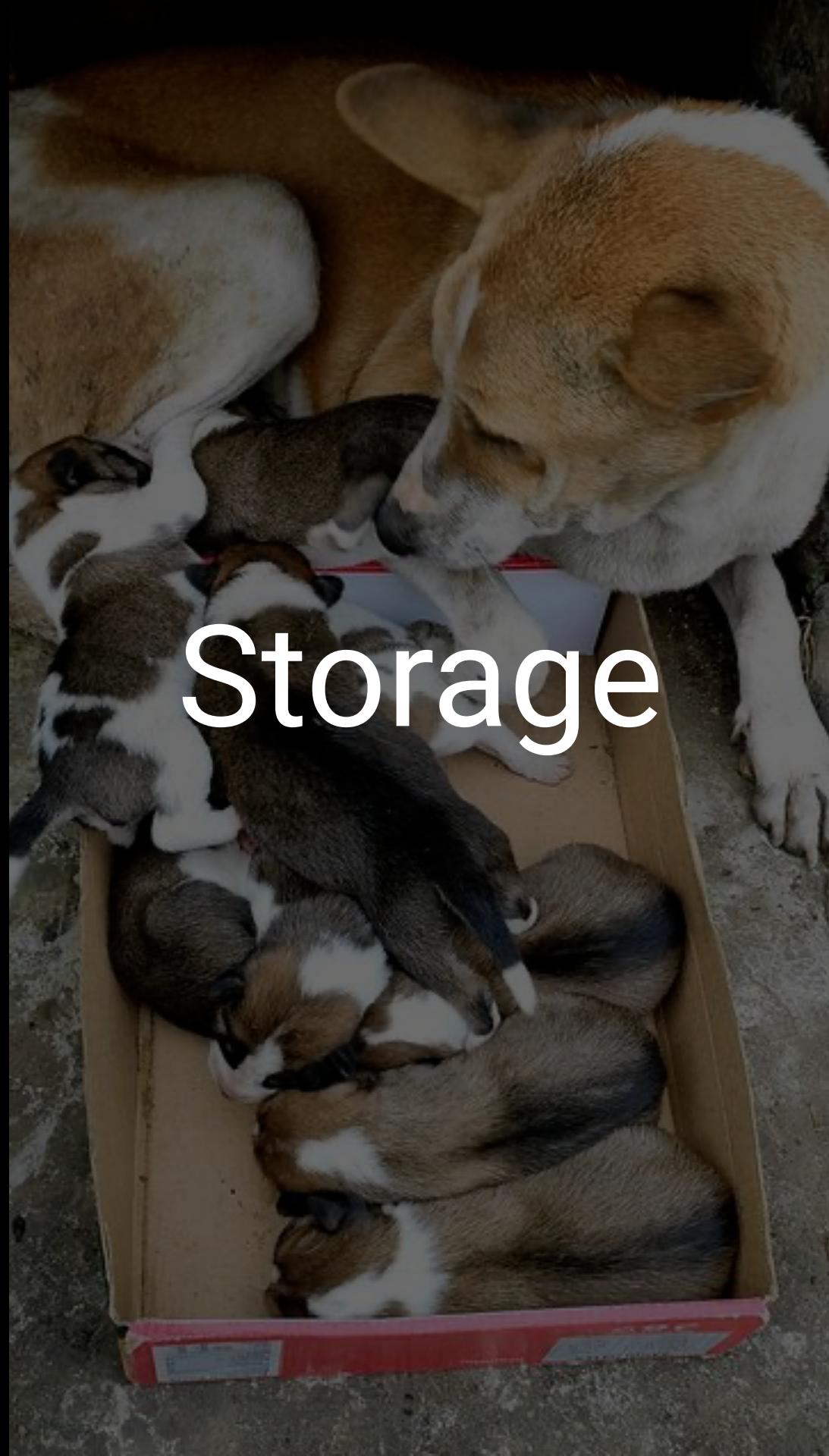
```
> update dogs set handle = '@itsdougthezug' where name = 'Franck';
```

ORA-00001: unique constraint (PDBADMIN.UNIQUE_HANDLE) violated on table PDBADMIN.DOGS columns (HANDLE)
ORA-03301: (ORA-00001 details) row with column values (HANDLE:'@itsdougthezug') already exists

```
> select name, handle from dogs order by handle;
```

NAME	HANDLE
Doug	@itsdougthezug
Marnie	@marniethedog
Bailey	[null]
Nala	[null]
Franck	[null]



A close-up photograph of a Corgi puppy sitting inside a small, open cardboard box. The puppy has a light brown and white coat. It is looking down at its front paws, which are resting on the edge of the box. The box is positioned on a dark, textured surface.

Storage

```
tab 0, row ?, @0x????  
tl: 24 fb: --H-FL-- lb: 0x2 cc: 3  
col 0: [ 4] 44 6f 75 67  
col 1: *NULL*  
col 2: [14] 40 69 74 73 64 6f 75 67 74 68 65 70 75 67
```

```
tab 0, row ?, @0x????  
tl: 16 fb: --H-FL-- lb: 0x1 cc: 2  
col 0: [ 4] 4e 61 6c 61  
col 1: [ 7] 4d 61 6e 75 65 6c 61
```

```
tab 0, row 1, @0x????  
tl: 10 fb: --H-FL-- lb: 0x1 cc: 1  
col 0: [ 6] 46 72 61 6e 63 6b
```

```
tab 0, row ?, @0x????  
tl: 24 fb: --H-FL-- lb: 0x2 cc: 3  
col 0: [ 4] 44 6f 75 67 <- name: Doug  
col 1: *NULL* <- owner: [null]  
col 2: [14] 40 69 74 73 64 6f 75 67 74 68 65 70 75 67 <- handle: @itsdougthe

ug



---


```

```
tab 0, row ?, @0x????  
tl: 16 fb: --H-FL-- lb: 0x1 cc: 2  
col 0: [ 4] 4e 61 6c 61 <- name: Nala  
col 1: [ 7] 4d 61 6e 75 65 6c 61 <- owner: Manuela  
                                         handle: [null]

---


```

```
tab 0, row ?, @0x????  
tl: 10 fb: --H-FL-- lb: 0x1 cc: 1  
col 0: [ 6] 46 72 61 6e 63 6b <- name: Franck  
                                         owner: [null]  
                                         handle: [null]
```

```
> select * from dogs;
```

NAME	OWNER	HANDLE
Nala	Manuela	[null] 16 \
Franck	[null]	[null] 10 \
Bailey	[null]	[null] 10 - 85
Doug	[null]	@itsdougthepug 24 /
Marnie	[null]	@marniethedog 25 /

```
> create table gods as select handle, owner, name from dogs;
```

Table GODS created.

```
> select * from gods;
```

HANDLE	OWNER	NAME
[null]	Manuela	Nala 17 \
[null]	[null]	Franck 12 \
[null]	[null]	Bailey 12 - 90
@itsdougthepug	[null]	Doug 24 /
@marniethedog	[null]	Marnie 25 /

```
> select * from dogs;
```

NAME	OWNER	HANDLE
Nala	Manuela	[null] 16 \
Franck	[null]	[null] 10 \
Bailey	[null]	[null] 10 - 85
Doug	[null]	@itsdougthepug 24 /
Marnie	[null]	@marniethedog 25 /

```
> create table gods as select handle, owner, name from dogs;
```

Table GODS created.

```
> select * from gods;
```

HANDLE	OWNER	NAME
[null]	Manuela	Nala 17 \
[null]	[null]	Franck 12 \
[null]	[null]	Bailey 12 - 90
@itsdougthepug	[null]	Doug 24 /
@marniethedog	[null]	Marnie 25 /

```
> select * from dogs;
```

NAME	OWNER	HANDLE
Nala	Manuela	[null] 16 \
Franck	[null]	[null] 10 \
Bailey	[null]	[null] 10 - 85
Doug	[null]	@itsdougthepug 24 /
Marnie	[null]	@marniethedog 25 /

```
> create table gods as select handle, owner, name from dogs;
```

Table GODS created.

```
> select * from gods;
```

HANDLE	OWNER	NAME
[null]	Manuela	Nala 17 \
[null]	[null]	Franck 12 \
[null]	[null]	Bailey 12 - 90
@itsdougthepug	[null]	Doug 24 /
@marniethedog	[null]	Marnie 25 /

A large brown dog, possibly a Dachshund or similar breed, stands on a light-colored sidewalk. The dog is facing towards the right of the frame. The background is dark and out of focus.

Fat Tables

```
> create table fat_dogs (
2      snack001 varchar2(16), snack002 varchar2(16), snack003 varchar2(16),
3      snack004 varchar2(16), snack005 varchar2(16), snack006 varchar2(16),
...
>52*      snack254 varchar2(16), snack255 varchar2(16), snack256 varchar2(16));
```

Table FAT_DOGS created.

```
> insert into fat_dogs(snack256) values ('Pupcorn');
```

1 row inserted.

```
> insert into fat_dogs(snack255) values ('Pawperoni');
```

1 row inserted.

```
> commit;
```

Commit complete

```
> create table fat_dogs (
2      snack001 varchar2(16), snack002 varchar2(16), snack003 varchar2(16),
3      snack004 varchar2(16), snack005 varchar2(16), snack006 varchar2(16),
...
>52*      snack254 varchar2(16), snack255 varchar2(16), snack256 varchar2(16));
```

Table FAT_DOGS created.

```
> insert into fat_dogs(snack256) values ('Pupcorn');
```

1 row inserted.

```
> insert into fat_dogs(snack255) values ('Pawperoni');
```

1 row inserted.

```
> commit;
```

Commit complete

```
tab 0, row ?, @0x????  
tl: 10 fb: --H-F--- lb: 0x1 cc: 1  
nrnid: 0x03c0089f.0  
col 0: *NULL*
```

```
tab 0, row ?, @0x????  
tl: 265 fb: -----L-- lb: 0x1 cc: 255  
col 0: *NULL*  
col 1: *NULL*  
...  
col 253: *NULL*  
col 254: [ 7] 50 75 70 63 6f 72 6e
```

```
tab 0, row ?, @0x????  
tl: 267 fb: --H-FL-- lb: 0x1 cc: 255  
col 0: *NULL*  
col 1: *NULL*  
...  
col 253: *NULL*  
col 254: [ 9] 50 61 77 70 65 72 6f 6e 69
```

```
tab 0, row ?, @0x????  
tl: 10 fb: --H-F--- lb: 0x1 cc: 1  
nrnid: 0x03c0089f.0  
col 0: *NULL* <- snack001: null
```

```
tab 0, row ?, @0x????  
tl: 265 fb: -----L-- lb: 0x1 cc: 255  
col 0: *NULL* <- snack002: null  
col 1: *NULL* <- snack003: null  
...  
col 253: *NULL* <- snack255: null  
col 254: [ 7] 50 75 70 63 6f 72 6e <- snack256: Pupcorn
```

```
tab 0, row ?, @0x????  
tl: 267 fb: --H-FL-- lb: 0x1 cc: 255  
col 0: *NULL* <- snack001: null  
col 1: *NULL* <- snack002: null  
...  
col 253: *NULL* <- snack254: null  
col 254: [ 9] 50 61 77 70 65 72 6f 6e 69 <- snack255: Pawperoni  
 <- snack256: null
```





```
> insert into people (name) values ('Patrick'),  
    ('Connor' );
```

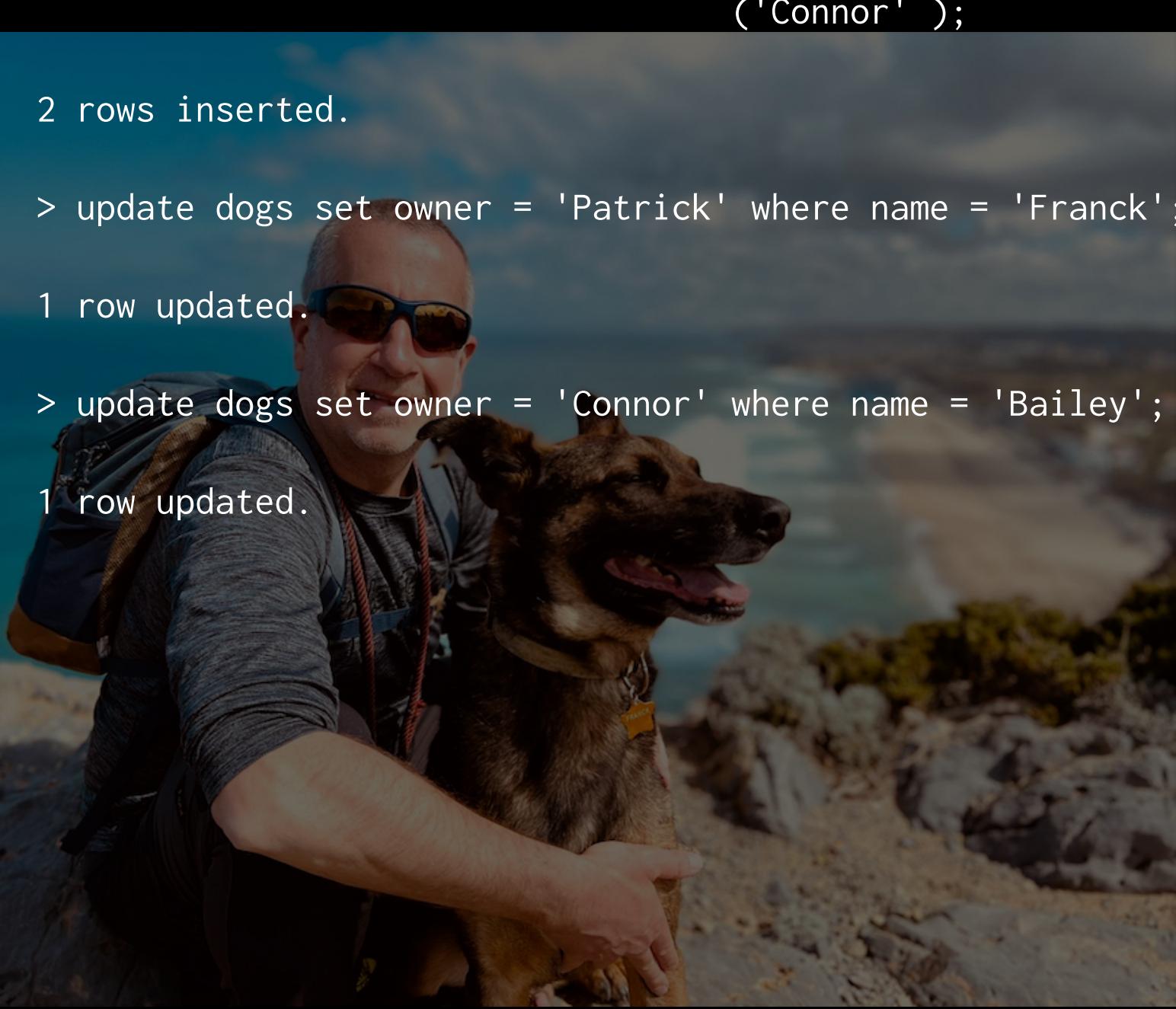
2 rows inserted.

```
> update dogs set owner = 'Patrick' where name = 'Franck';
```

1 row updated.

```
> update dogs set owner = 'Connor' where name = 'Bailey';
```

1 row updated.



Statistics

```
> select owner, count(*) from dogs group by owner order by owner;
```

OWNER	COUNT(*)
Connor	1
Manuela	1
Patrick	1
[null]	2

```
> exec dbms_stats.gather_table_stats(null, 'dogs');
```

PL/SQL procedure successfully completed.

```
> select num_nulls, num_distinct from user_tab_cols where table_name = 'DOGS'  
2*                                and column_name = 'OWNER';
```

NUM_NULLS	NUM_DISTINCT
2	3

```
> select owner, count(*) from dogs group by owner order by owner;
```

OWNER	COUNT(*)
Connor	1
Manuela	1
Patrick	1
[null]	2

```
> exec dbms_stats.gather_table_stats(null, 'dogs');
```

PL/SQL procedure successfully completed.

```
> select num_nulls, num_distinct from user_tab_cols where table_name = 'DOGS'  
2*                                and column_name = 'OWNER';
```

NUM_NULLS	NUM_DISTINCT
2	3

```
> select owner, count(*) from dogs group by owner order by owner;
```

OWNER	COUNT(*)
Connor	1
Manuela	1
Patrick	1
[null]	2

```
> exec dbms_stats.gather_table_stats(null, 'dogs');
```

PL/SQL procedure successfully completed.

```
> select num_nulls, num_distinct from user_tab_cols where table_name = 'DOGS'  
2*                                and column_name = 'OWNER';
```

NUM_NULLS	NUM_DISTINCT
2	3

```
> select name from dogs where owner is null;
```

NAME

Doug

Marnie

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1		2	6
* 1	TABLE ACCESS FULL	DOGS	1	2	2	6

Predicate Information (identified by operation id):

1 - filter("OWNER" IS NULL)

```
> select name from dogs where owner is null;
```

NAME

Doug

Marnie

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1		2	6
* 1	TABLE ACCESS FULL	DOGS	1	2	2	6

Predicate Information (identified by operation id):

1 - filter("OWNER" IS NULL)

Indexes

```
> select owner, count(*) from dogs group by owner order by owner;
```

OWNER	COUNT(*)
Connor	1
Manuela	1
Patrick	1
[null]	2

```
> create index dogs_owner on dogs(owner);
```

Index DOGS_OWNER created.

```
> select num_rows, distinct_keys from user_indexes where index_name = 'DOGS_OWNER';
```

NUM_ROWS	DISTINCT_KEYS
3	3

```
> select owner, count(*) from dogs group by owner order by owner;
```

OWNER	COUNT(*)
Connor	1
Manuela	1
Patrick	1
[null]	2

```
> create index dogs_owner on dogs(owner);
```

Index DOGS_OWNER created.

```
> select num_rows, distinct_keys from user_indexes where index_name = 'DOGS_OWNER';
```

NUM_ROWS	DISTINCT_KEYS
3	3

```
> select owner, count(*) from dogs group by owner order by owner;
```

OWNER	COUNT(*)
Connor	1
Manuela	1
Patrick	1
[null]	2

```
> create index dogs_owner on dogs(owner);
```

Index DOGS_OWNER created.

```
> select num_rows, distinct_keys from user_indexes where index_name = 'DOGS_OWNER';
```

NUM_ROWS	DISTINCT_KEYS
3	3

```
> select /*+ index_rs_asc(dogs) */ name from dogs where owner is null;
```

NAME

Doug

Marnie

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		2	00:00:00.01	6
* 1	TABLE ACCESS FULL	DOGS	1	2	2	00:00:00.01	6

Predicate Information (identified by operation id):

1 - filter("OWNER" IS NULL)

```
> select /*+ index_rs_asc(dogs) */ name from dogs where owner is null;
```

NAME

Doug

Marnie

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1	1	2	00:00:00.01	6
* 1	TABLE ACCESS FULL	DOGS	1	2	2	00:00:00.01	6

Predicate Information (identified by operation id):

1 - filter("OWNER" IS NULL)

```
> select owner, name from dogs order by owner, name;
```

OWNER	NAME
Connor	Bailey
Manuela	Nala
Patrick	Franck
[null]	Doug
[null]	Marnie

```
> create index dogs_owner_name on dogs(owner, name);
```

Index DOGS_OWNER_NAME created.

```
> select num_rows, distinct_keys from user_indexes where index_name = 'DOGS_OWNER_NAME';
```

NUM_ROWS	DISTINCT_KEYS
5	5

```
> select owner, name from dogs order by owner, name;
```

OWNER	NAME
Connor	Bailey
Manuela	Nala
Patrick	Franck
[null]	Doug
[null]	Marnie

```
> create index dogs_owner_name on dogs(owner, name);
```

Index DOGS_OWNER_NAME created.

```
> select num_rows, distinct_keys from user_indexes where index_name = 'DOGS_OWNER_NAME';
```

NUM_ROWS	DISTINCT_KEYS
5	5

```
> select owner, name from dogs order by owner, name;
```

OWNER	NAME
Connor	Bailey
Manuela	Nala
Patrick	Franck
[null]	Doug
[null]	Marnie

```
> create index dogs_owner_name on dogs(owner, name);
```

Index DOGS_OWNER_NAME created.

```
> select num_rows, distinct_keys from user_indexes where index_name = 'DOGS_OWNER_NAME';
```

NUM_ROWS	DISTINCT_KEYS
5	5

```
> select name from dogs where owner is null;
```

NAME

Doug

Marnie

Id	Operation	Name	Starts	E-Rows	A-Rows	Buffers
0	SELECT STATEMENT		1		2	1
* 1	INDEX RANGE SCAN	DOGS_OWNER_NAME	1	2	2	1

Predicate Information (identified by operation id):

1 - access("OWNER" IS NULL)

```
> select name from dogs where owner is null;
```

NAME

Doug

Marnie

	Id	Operation		Name		Starts		E-Rows		A-Rows		Buffers	
	0	SELECT STATEMENT				1				2		1	
*	1	INDEX RANGE SCAN		DOGS_OWNER_NAME		1		2		2		1	

Predicate Information (identified by operation id):

1 - access("OWNER" IS NULL)

I HAVEN'T EATEN IN AN HOUR



THAT'S 7 DOG HOURS

```
> select name, status from dogs;
```

NAME	STATUS
Nala	Full
Franck	Hungry
Bailey	Full
Doug	Full
Marnie	Full

```
> alter table dogs add is_hungry number generated always as (
  2   case
  3     when status = 'Hungry' then 1
  4     else null
  5   end
  6*) virtual;
```

Table DOGS altered.

```
> select name, status, is_hungry from dogs;
```

NAME	STATUS	IS_HUNGRY
Nala	Full	[null]
Franck	Hungry	1
Bailey	Full	[null]
Doug	Full	[null]
Marnie	Full	[null]

```
> alter table dogs add is_hungry number generated always as (
  2   case
  3     when status = 'Hungry' then 1
  4     else null
  5   end
  6* ) virtual;
```

Table DOGS altered.

```
> select name, status, is_hungry from dogs;
```

NAME	STATUS	IS_HUNGRY
Nala	Full	[null]
Franck	Hungry	1
Bailey	Full	[null]
Doug	Full	[null]
Marnie	Full	[null]

```
> create index hungry_dogs on dogs(is_hungry);
```

Index HUNGRY_DOGS created.

```
> select index_type, num_rows, distinct_keys from user_indexes where index_name = 'HUNGRY_DOGS';
```

INDEX_TYPE	NUM_ROWS	DISTINCT_KEYS
FUNCTION-BASED NORMAL	1	1

```
> exec dbms_stats.gather_table_stats(null, 'dogs');
```

PL/SQL procedure successfully completed.

```
> create index hungry_dogs on dogs(is_hungry);
```

Index HUNGRY_DOGS created.

```
> select index_type, num_rows, distinct_keys from user_indexes where index_name = 'HUNGRY_DOGS';
```

INDEX_TYPE	NUM_ROWS	DISTINCT_KEYS
FUNCTION-BASED NORMAL	1	1

```
> exec dbms_stats.gather_table_stats(null, 'dogs');
```

PL/SQL procedure successfully completed.

```
> create index hungry_dogs on dogs(is_hungry);
```

Index HUNGRY_DOGS created.

```
> select index_type, num_rows, distinct_keys from user_indexes where index_name = 'HUNGRY_DOGS';
```

INDEX_TYPE	NUM_ROWS	DISTINCT_KEYS
FUNCTION-BASED NORMAL	1	1

```
> exec dbms_stats.gather_table_stats(null, 'dogs');
```

PL/SQL procedure successfully completed.

```
> select name from dogs where is_hungry = 1;
```

NAME

Franck

```
> select * from dbms_xplan.display_cursor();
```

Id	Operation	Name	Rows
0	SELECT STATEMENT		
1	TABLE ACCESS BY INDEX ROWID BATCHED	DOGS	1
* 2	INDEX RANGE SCAN	HUNGRY_DOGS	1

Predicate Information (identified by operation id):

2 - access("IS_HUNGRY"=1)

```
> select name from dogs where is_hungry = 1;
```

NAME

Franck

```
> select * from dbms_xplan.display_cursor();
```

Id	Operation	Name	Rows	
0	SELECT STATEMENT			
1	TABLE ACCESS BY INDEX ROWID BATCHED	DOGS	1	
* 2	INDEX RANGE SCAN	HUNGRY_DOGS	1	

Predicate Information (identified by operation id):

2 - access("IS_HUNGRY"=1)



WHAT DOES NULL SIGNIFY?



I DON'T KNOW



THAT'S RIGHT



.....