

Tailoring CREATE PUBLICATION

a complete syntax guide

Peter Smith

- Fujitsu Open-Source Software Team
- Fujitsu Australia



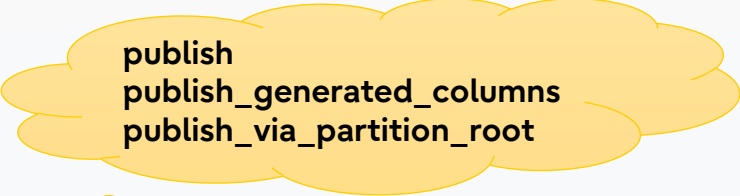
Agenda

- Intro / Recap LR
- Syntax Knobs
- Combos and flexibility
- Things I didn't talk about
- Future
- Summary
- References
- Q/A ← at the end



Intro

```
CREATE PUBLICATION name
[ FOR ALL TABLES |
  FOR publication_object [, ...] ]
[ WITH ( publication_parameter [= value] [, ...] ) ]
```



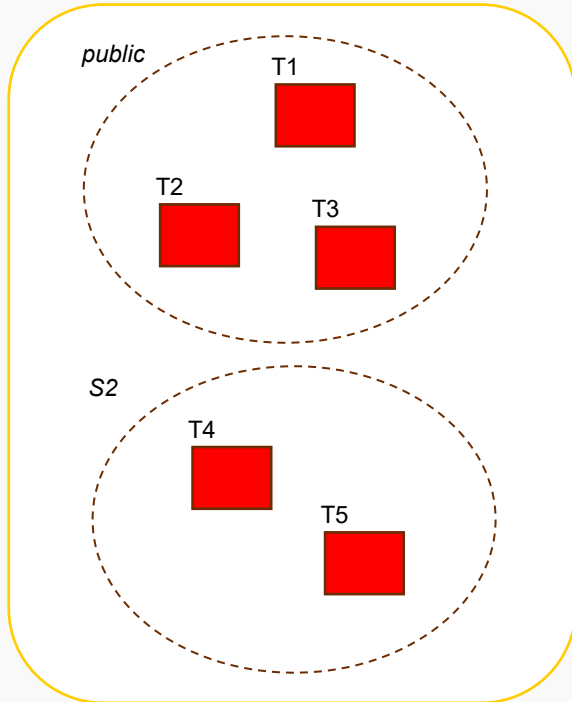
publish
publish_generated_columns
publish_via_partition_root

where *publication_object* is one of:

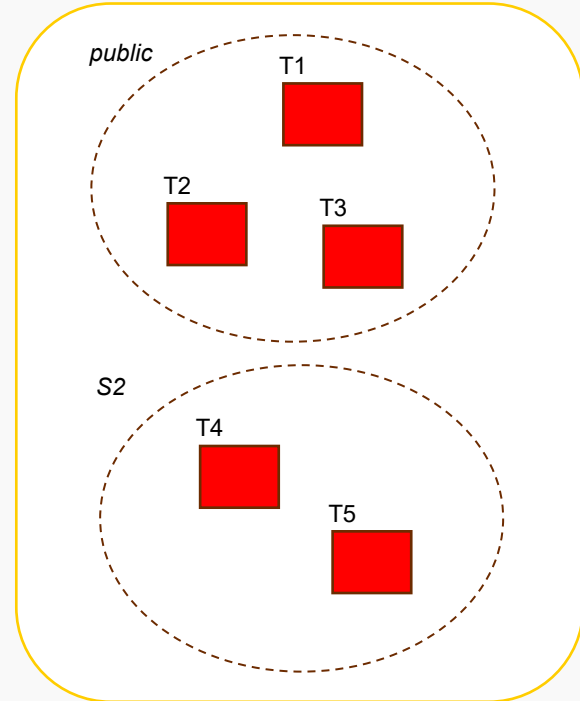
```
TABLE { [ ONLY ] table_name [ * ] [ ( column_name [, ...] ) ]
      [ WHERE ( expression ) ] } [, ...]
TABLES IN SCHEMA { schema_name | CURRENT_SCHEMA } [, ...]
```

Recap – Logical Replication

CREATE PUBLICATION *mypub* ...

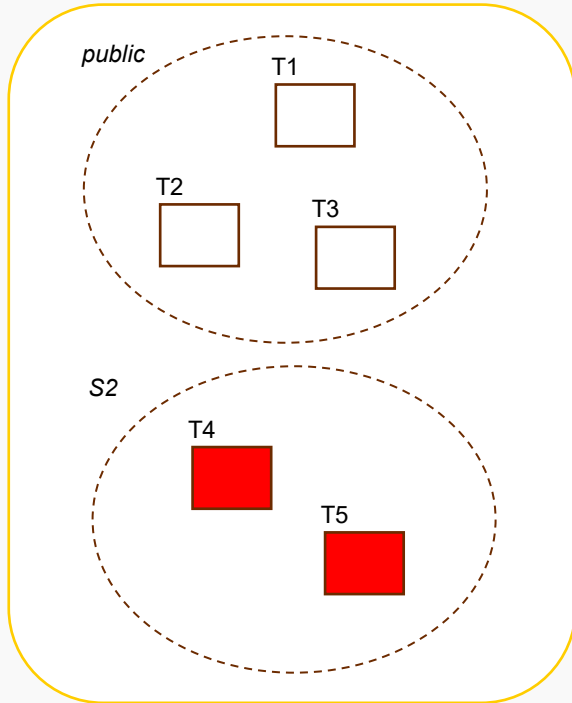


CREATE SUBSCRIPTION *mysub* ...
PUBLICATION *mypub* ...

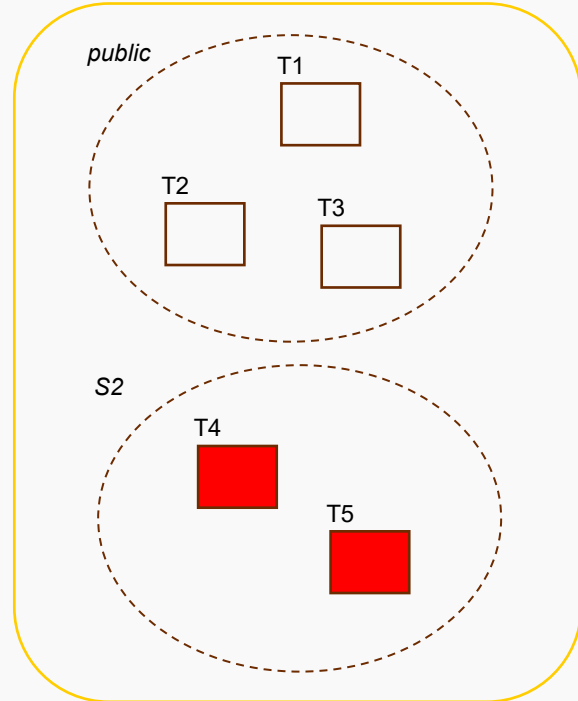


Recap – Logical Replication

CREATE PUBLICATION *mypub* ...

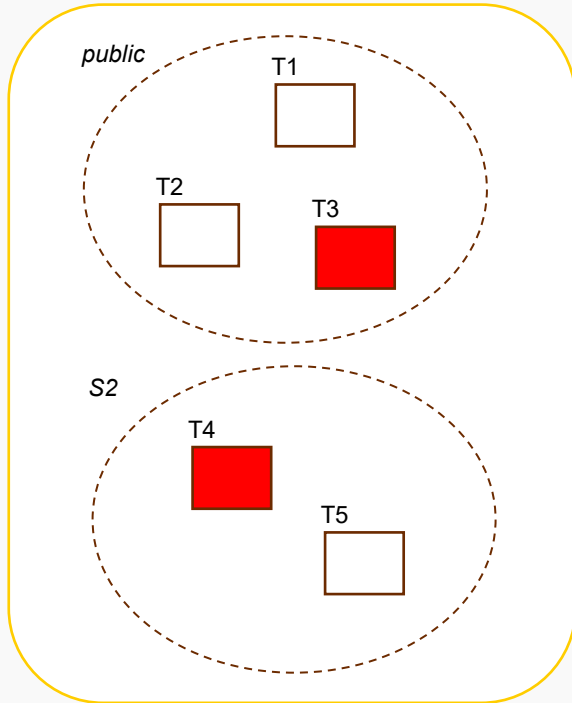


CREATE SUBSCRIPTION *mysub* ...
PUBLICATION *mypub* ...

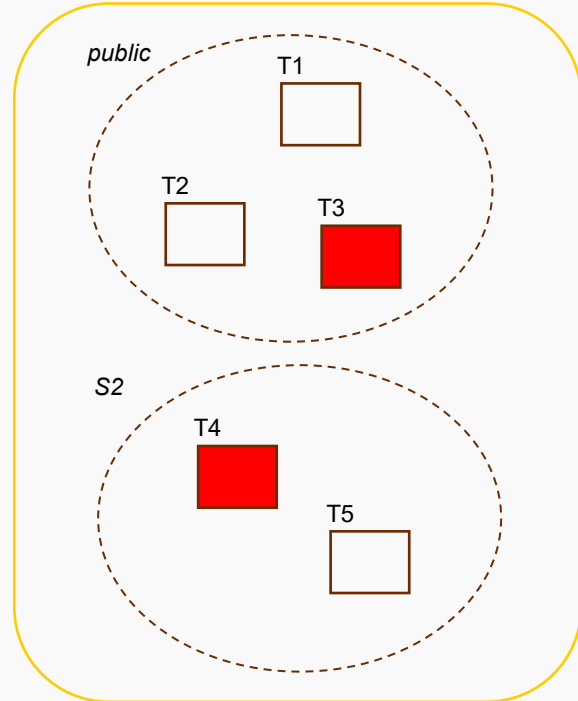


Recap – Logical Replication

CREATE PUBLICATION *mypub* ...

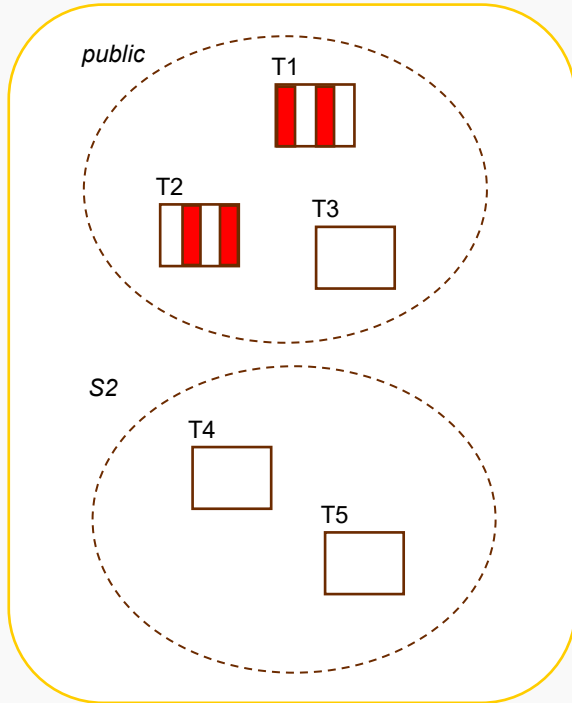


CREATE SUBSCRIPTION *mysub* ...
PUBLICATION *mypub* ...

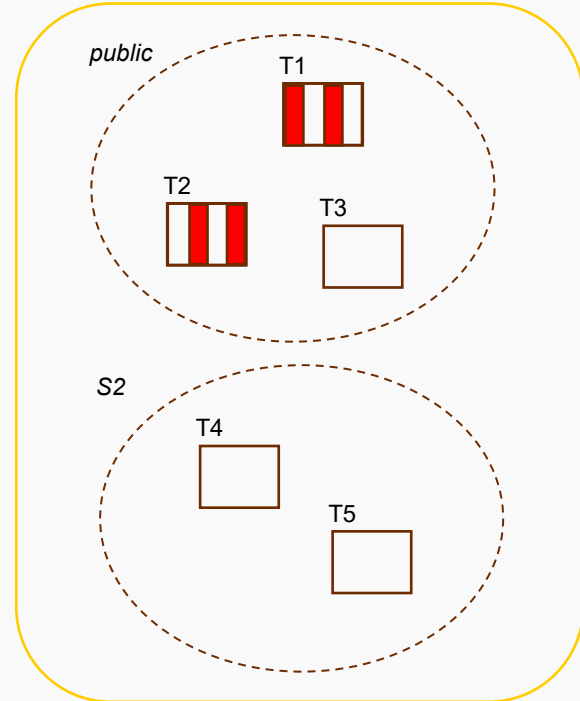


Recap – Logical Replication

CREATE PUBLICATION *mypub* ...

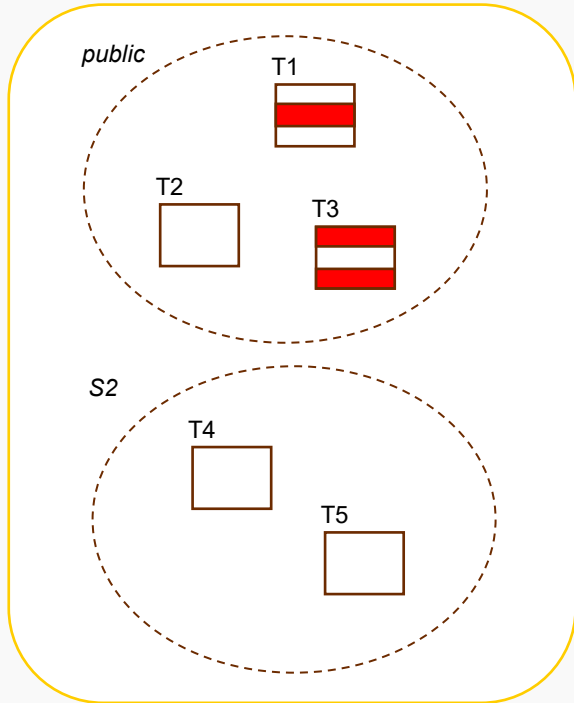


CREATE SUBSCRIPTION *mysub* ...
PUBLICATION *mypub* ...

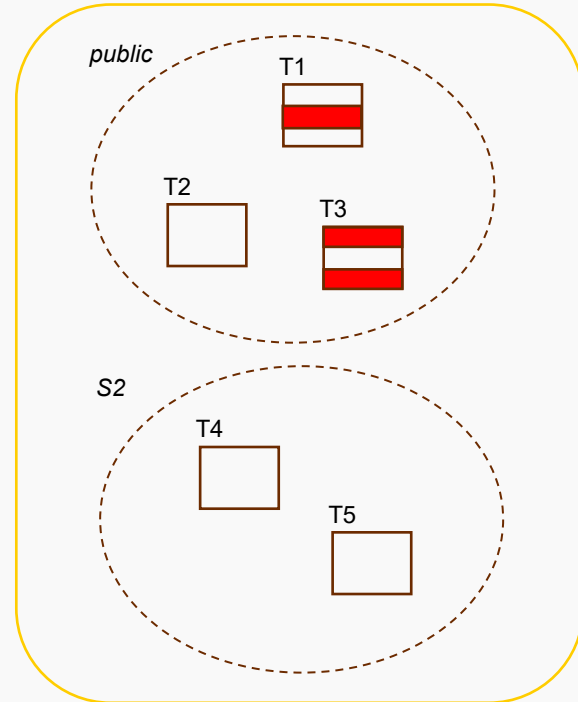


Recap – Logical Replication

CREATE PUBLICATION *mypub* ...

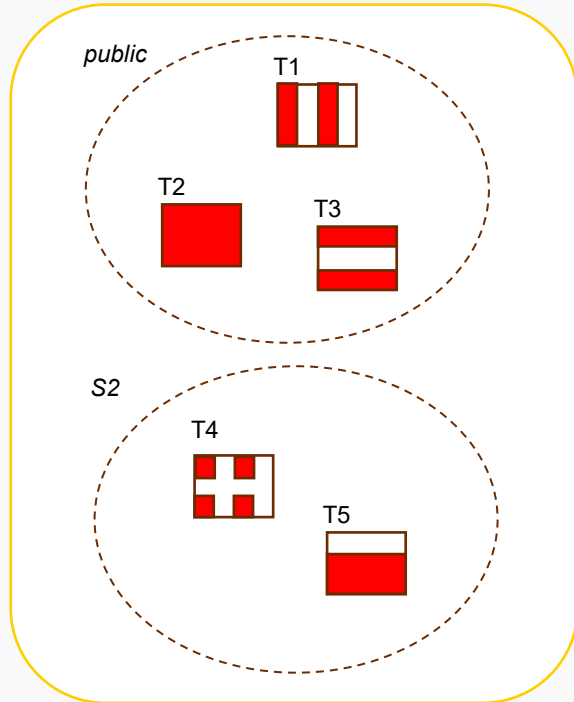


CREATE SUBSCRIPTION *mysub* ...
PUBLICATION *mypub* ...

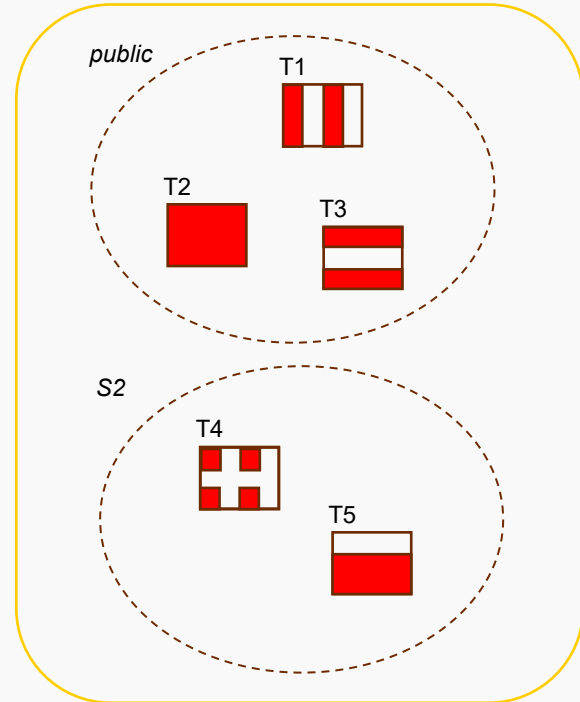


Recap – Logical Replication

CREATE PUBLICATION *mypub* ...



CREATE SUBSCRIPTION *mysub* ...
PUBLICATION *mypub* ...



Recap – Logical Replication

CREATE PUBLICATION *mypub* ...

TA

c1	c2	c3	c4

TB

c1	c2	c3	c4	c5

TC

c1	c2	c3



CREATE SUBSCRIPTION *mysub* ...
PUBLICATION *mypub* ...

TA

c1	c2	c3	c4

TB

c1	c2	c3

TC

c1	c2	c3	c4	c5

Table
shapes

Catalogs, Views, and psql Describe

```
psql# SELECT * FROM pg_publication
```

```
psql# SELECT * FROM pg_publication_namespace
```

```
psql# SELECT * FROM pg_publication_rel
```

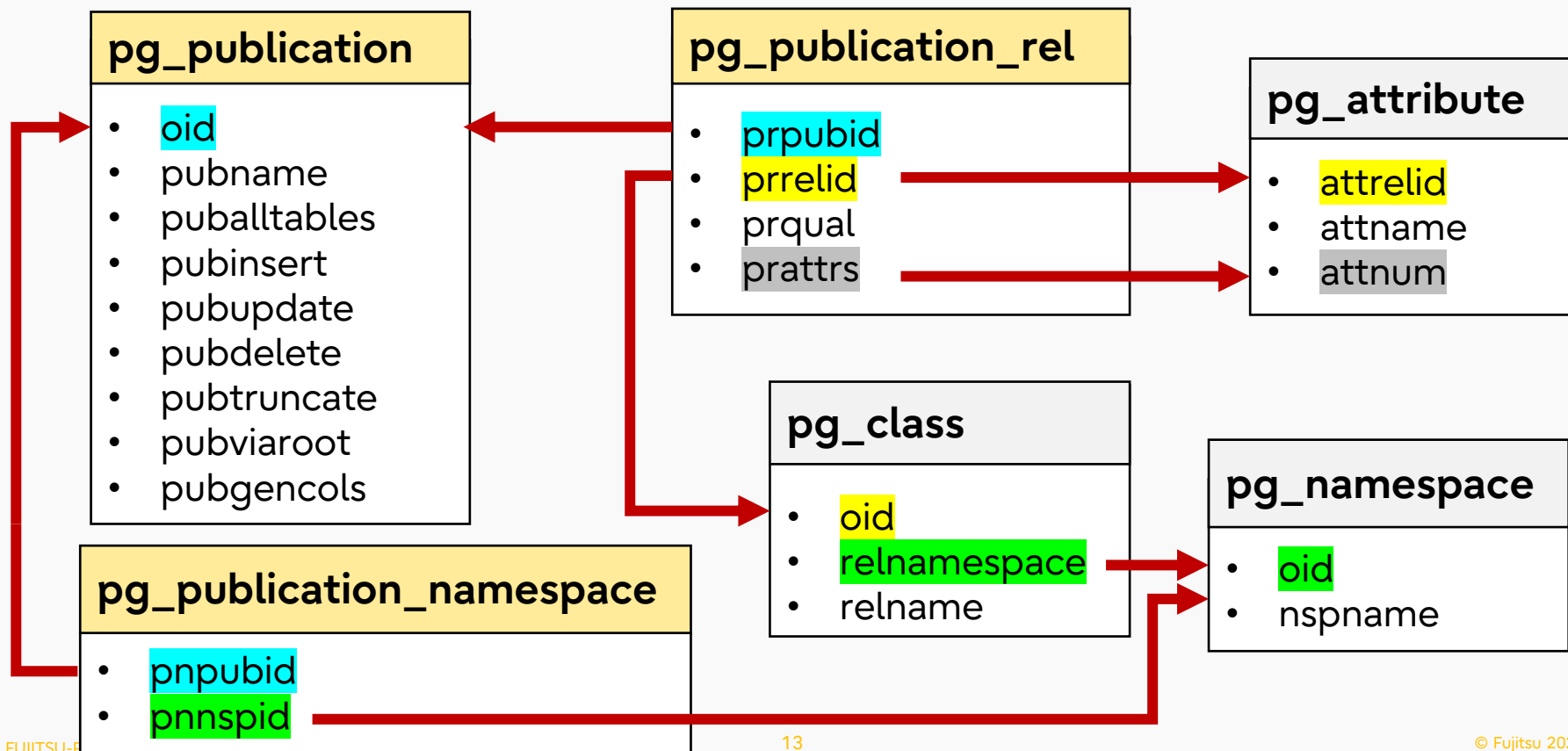
```
psql# SELECT * FROM pg_publication_tables
```

```
psql# \dRp+ mypub
```

```
psql# \d mytab . . .
```



Shows publications
involving mytab



CREATE PUBLICATION ...

Everything



Nothing

```
pub_18 ... WITH (publish_generated_columns=stored)
pub_13 FOR ALL TABLES
pub_15 FOR TABLES IN SCHEMA s1,s2
pub_13 FOR TABLE t1,t2,t3
pub_15 FOR TABLE t1 WHERE (x=99)
pub_15 FOR TABLE t1(c1,c2,c3)
pub_13 ... WITH (publish=...)
pub_13
```

CREATE PUBLICATION ...

Everything

```
pub_18 ... WITH (publish_generated_columns=stored)
```

```
pub_13 FOR ALL TABLES
```

```
pub_15 FOR TABLES IN SCHEMA s1,s2
```

```
pub_13 FOR TABLE t1,t2,t3
```

```
pub_15 FOR TABLE t1 WHERE (x=99)
```

```
pub_15 FOR TABLE t1 (c1,c2,c3)
```

```
pub_13 ... WITH (publish=...)
```

```
pub_13
```

Nothing

CREATE PUBLICATION ...

Everything



Nothing

```
pub_18 ... WITH (publish_generated_columns=stored)
pub_13 FOR ALL TABLES
pub_15 FOR TABLES IN SCHEMA s1,s2
pub_13 FOR TABLE t1,t2,t3
pub_15 FOR TABLE t1 WHERE (x=99)
pub_15 FOR TABLE t1 (c1,c2,c3)
pub_13 ... WITH (publish=...)
pub_13
```

CREATE PUBLICATION ...

Everything

pub_18 ... WITH (publish_generated_columns=stored)

pub_13 FOR ALL TABLES

pub_15 FOR TABLES IN SCHEMA s1,s2

pub_13 FOR TABLE t1,t2,t3

pub_15 FOR TABLE t1 WHERE (x=99)

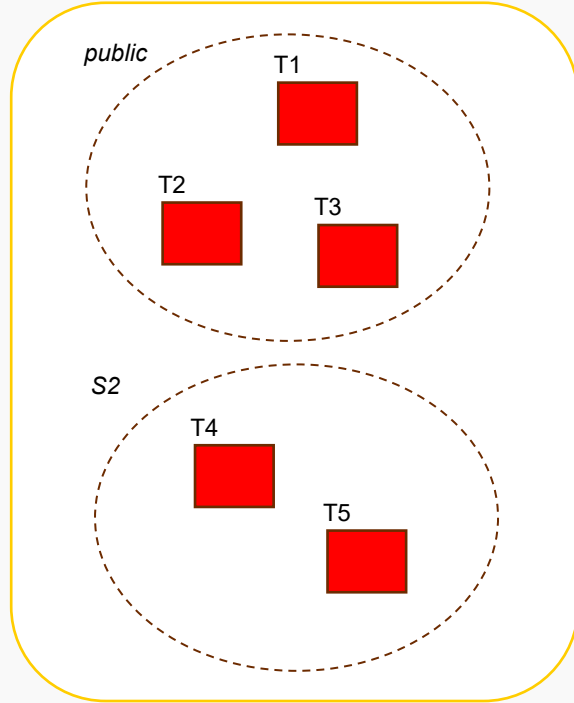
pub_15 FOR TABLE t1(c1,c2,c3)

pub_13 ... WITH (publish=...)

pub_13

Nothing

Syntax Knobs



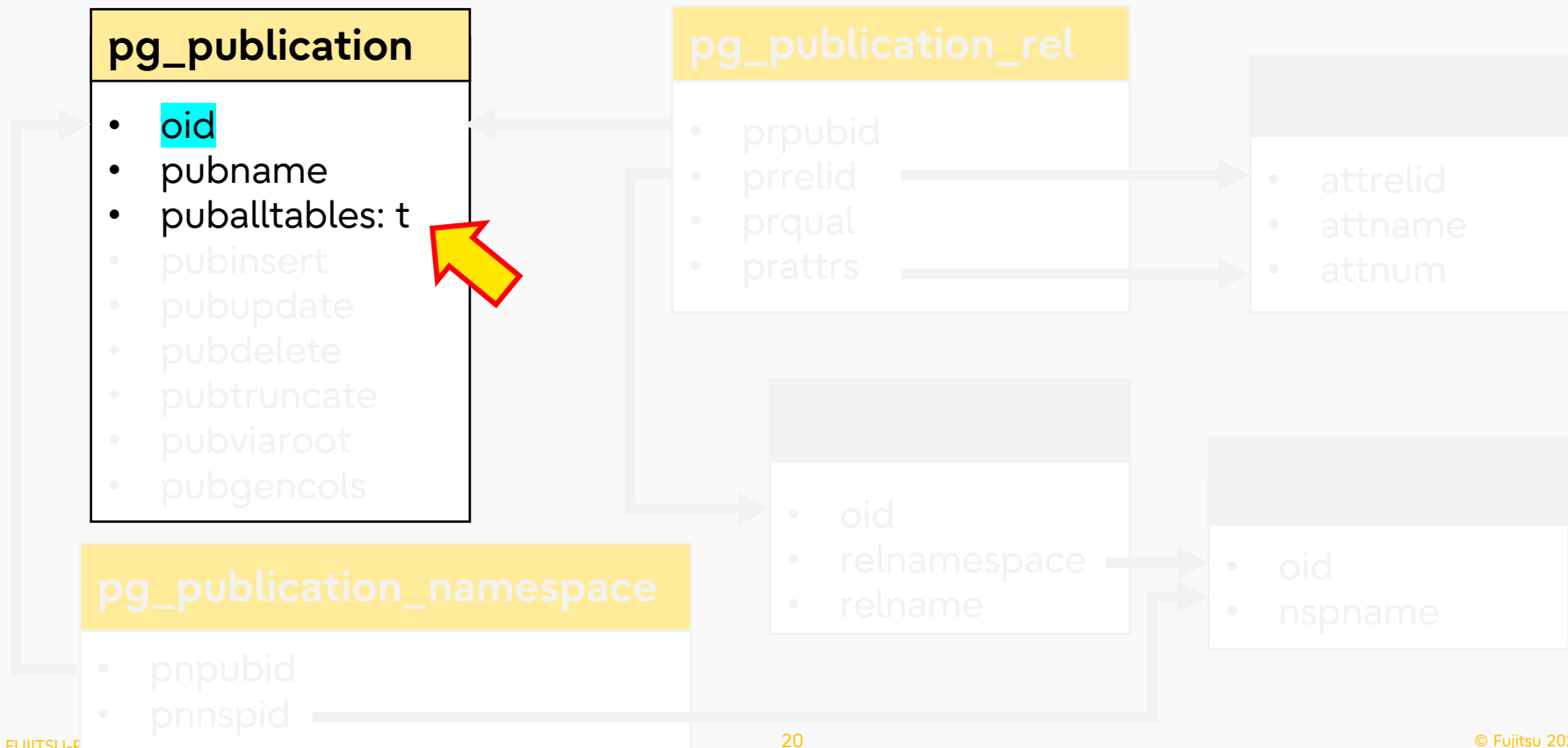
CREATE PUBLICATION `pub_for_all_tables` FOR ALL TABLES;

Since PG13.

Publish:

- All tables (present and future) from all schemas
- All data from those tables (excluding generated columns)

Future
tables?



```
test_pub=# \dRp+ pub_for_all_tables
```

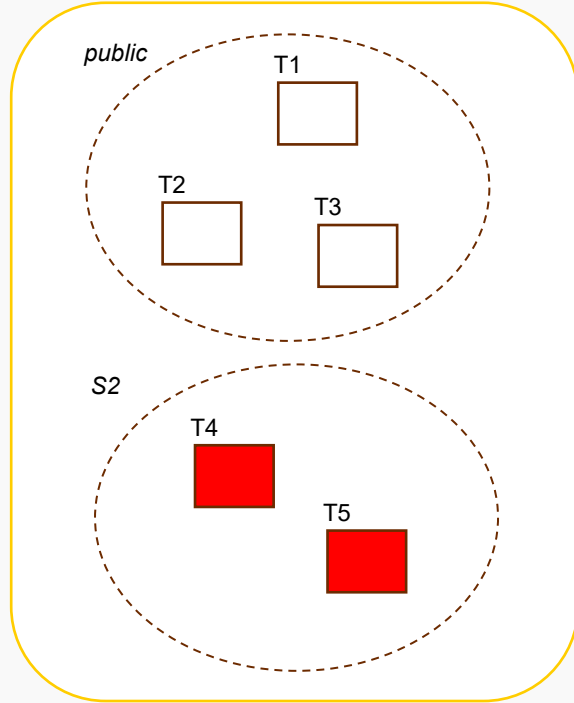
Publication pub_for_all_tables							
Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	t	t	t	t	t	none	f

(1 row)

```
test_pub=# SELECT * FROM pg_publication_tables;
```

pubname	schemaname	tablename	attnames	rowfilter
pub_for_all_tables	public	t1	{c1,c2,c3,c4}	
pub_for_all_tables	public	t2	{c1,c2,c3,c4}	
pub_for_all_tables	public	t3	{c1,c2,c3,c4}	
pub_for_all_tables	s2	t4	{c1,c2,c3,c4}	
pub_for_all_tables	s2	t5	{c1,c2,c3,c4}	

(5 rows)



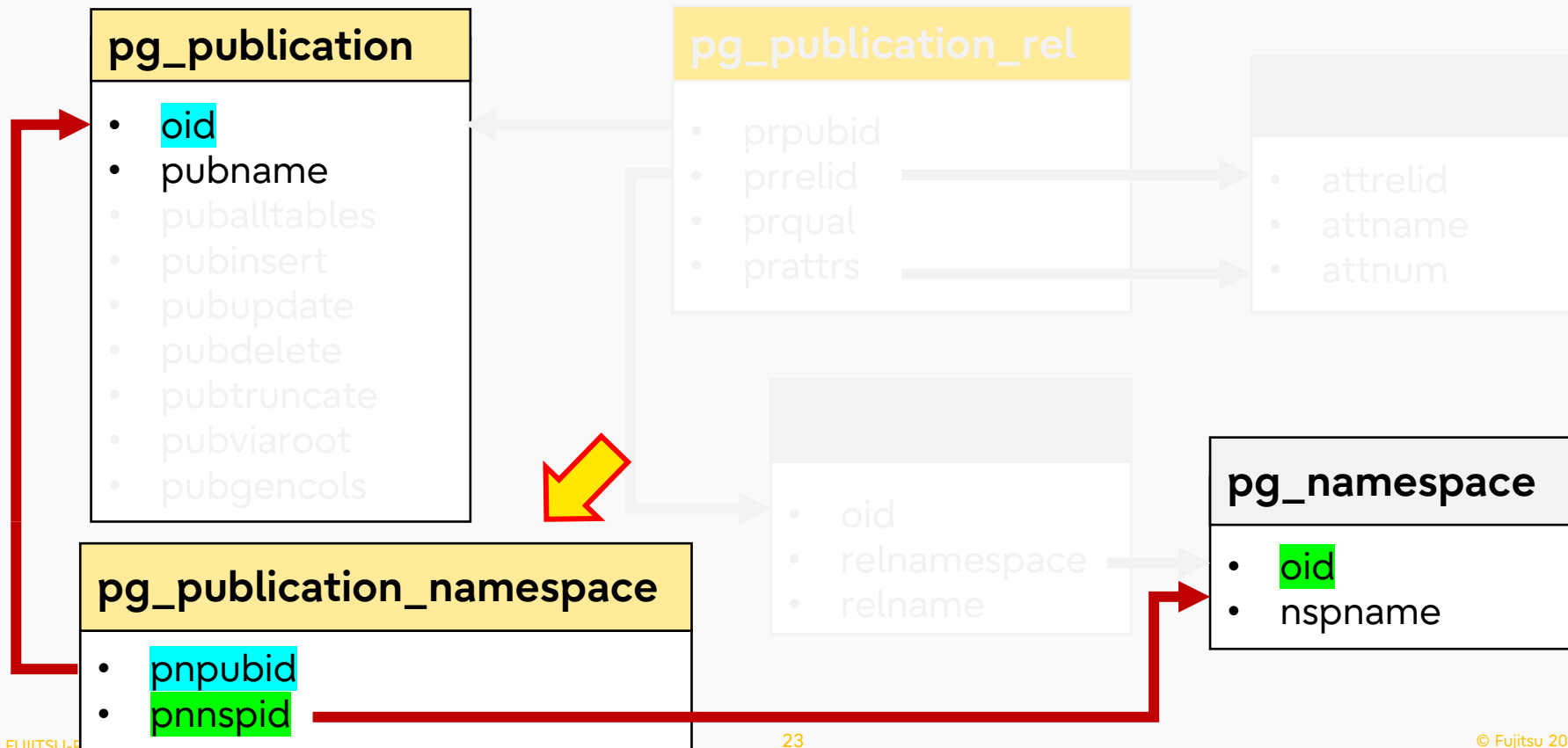
CREATE PUBLICATION *pub_for_schema*
FOR TABLES IN SCHEMA *s2*;

Since PG15

Publish:

- All tables (present and future) from the nominated schema
- All data from those tables (excluding generated columns)

Future
tables...



```
test_pub=# \dRp+ pub_for_schema
```

Publication pub_for_schema

Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	t	t	t	t	none	f

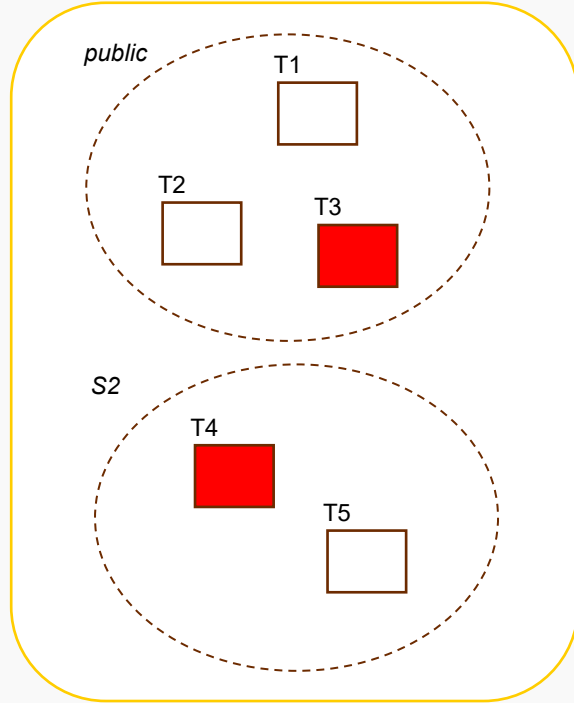
```
Tables from schemas:
```

```
"s2"
```

```
test_pub=# SELECT * FROM pg_publication_tables;
```

pubname	schemaname	tablename	attnames	rowfilter
pub_for_schema	s2	t4	{c1,c2,c3,c4}	
pub_for_schema	s2	t5	{c1,c2,c3,c4}	

(2 rows)



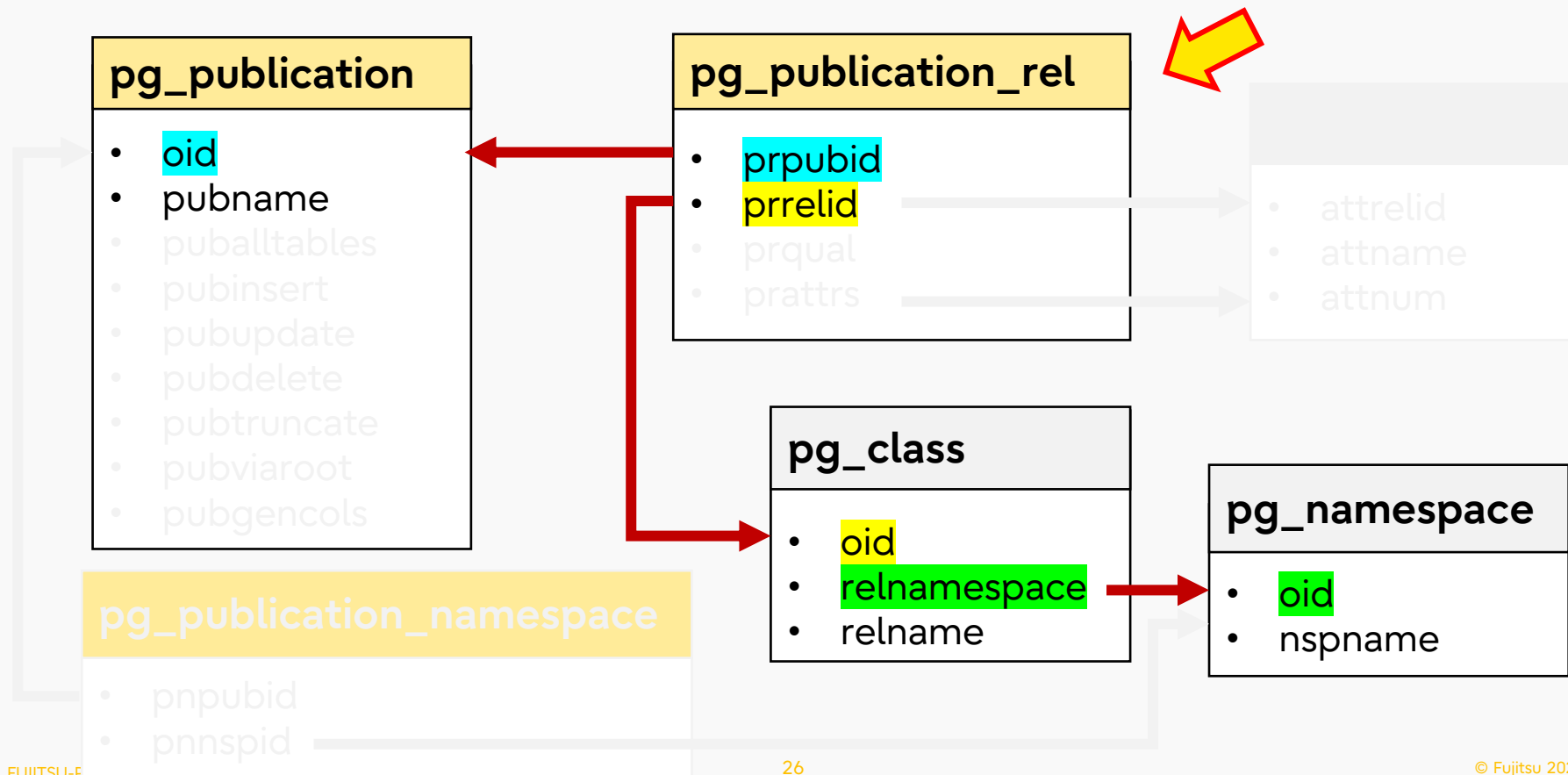
CREATE PUBLICATION `pub_for_table`
FOR TABLE `t3`, `s2.t4`;

Since PG13

Publish:

- Only the specified tables
- All data from those tables (excluding generated column data)

Specific Tables



```
test_pub=# \dRp+
```

Publication pub_for_table

Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	t	t	t	t	none	f

Tables:

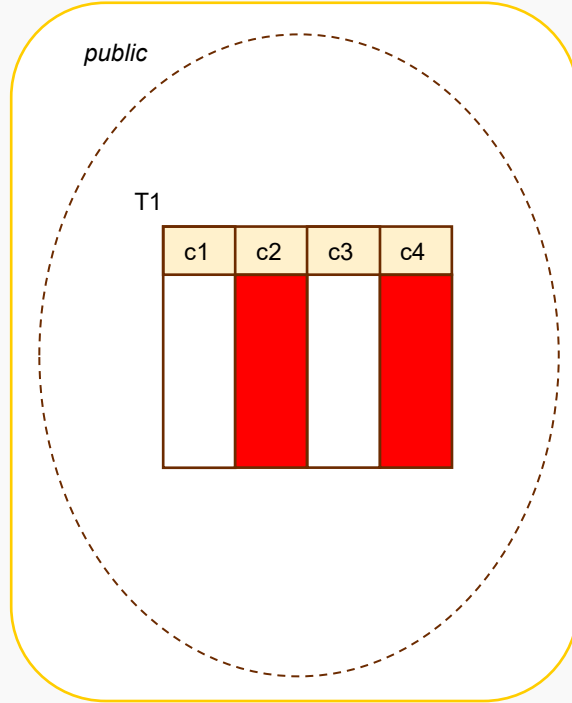
"public.t3"

"s2.t4"

```
test_pub=# SELECT * FROM pg_publication_tables;
```

pubname	schemaname	tablename	attnames	rowfilter
pub_for_table	public	t3	{c1,c2,c3,c4}	
pub_for_table	s2	t4	{c1,c2,c3,c4}	

(2 rows)



CREATE PUBLICATION `pub_collist`
FOR TABLE `t1(c2, c4);`

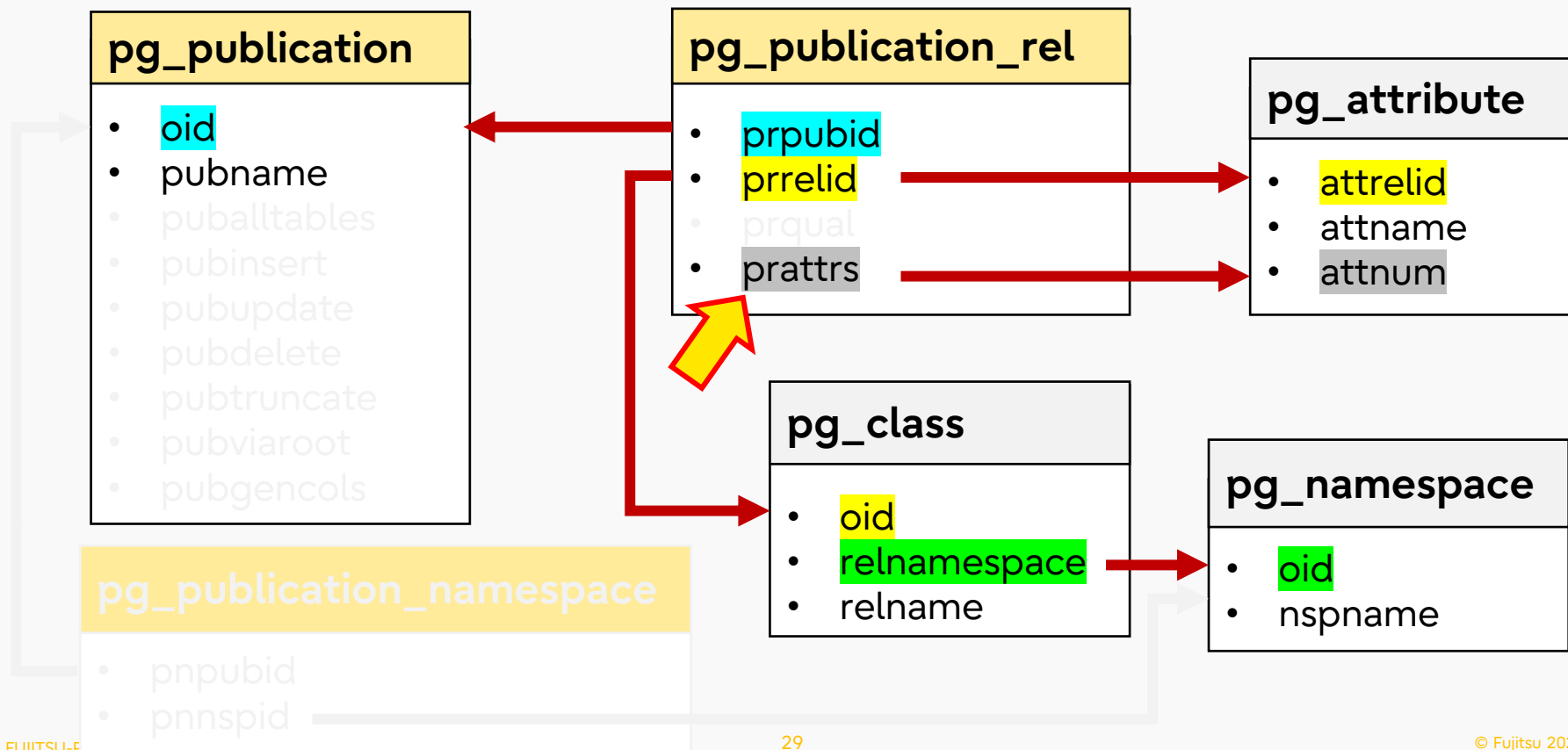
Since PG15

Publish:

- Only the specified tables
- Only data from the specified columns

Why?

Sensitive data
Subscriber table cols



```
test_pub=# \dRp+
```

Publication pub_collist

Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	t	t	t	t	none	f

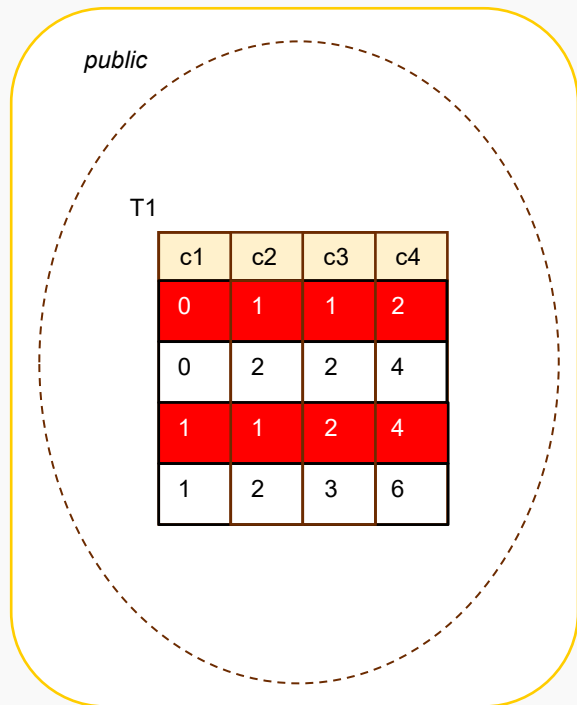
Tables:

"public.t1" (c2, c4)

```
test_pub=# SELECT * FROM pg_publication_tables;
```

pubname	schemaname	tablename	attnames	rowfilter
pub_collist	public	t1	{c2,c4}	

(1 row)



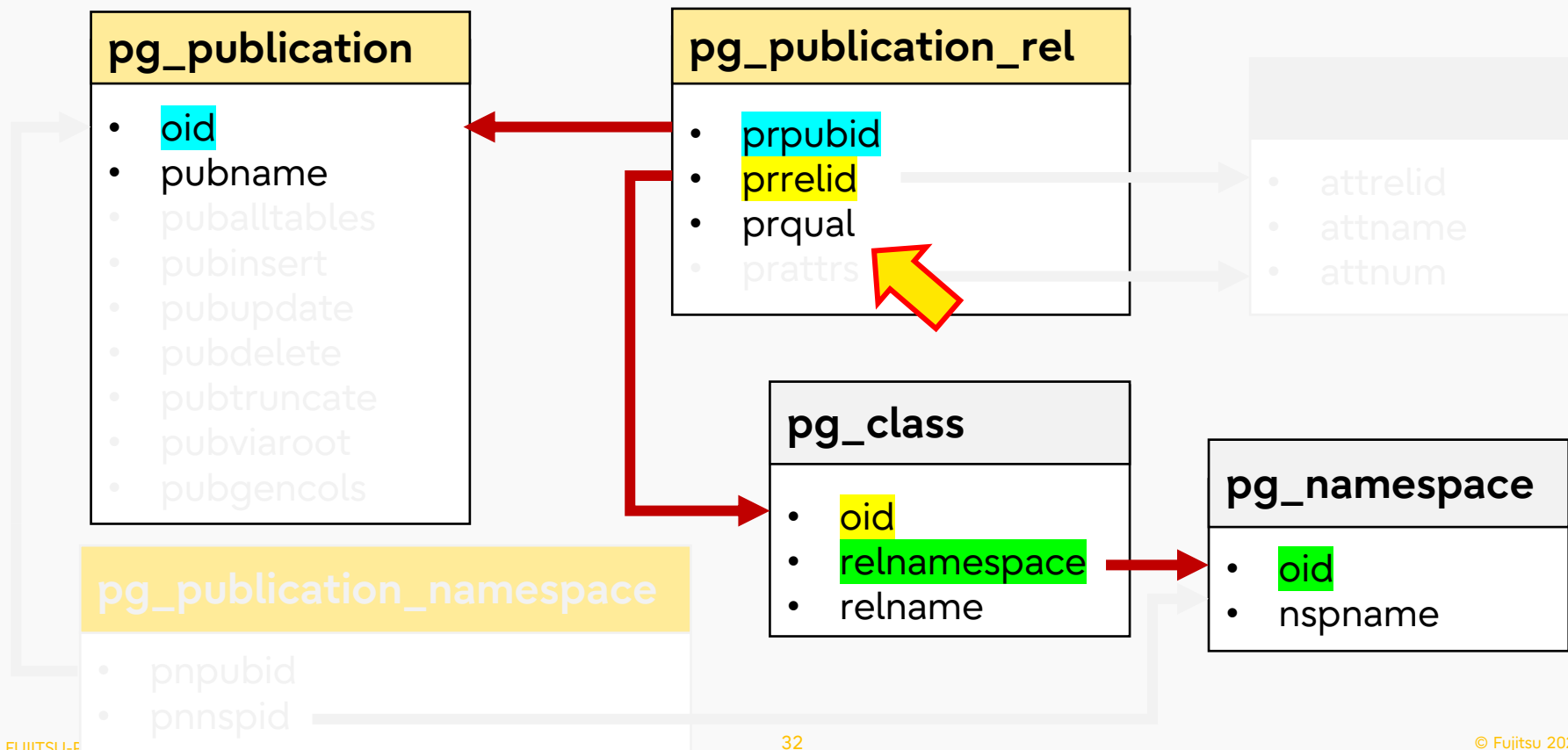
CREATE PUBLICATION `pub_rowfilter`
FOR TABLE `t1` WHERE (`c2 = 1`);

Since PG15

Publish:

- Only the specified tables
- Only data from rows matching the specified condition

Row Filtering



```
test_pub=# \dRp+
```

Publication pub_rowfilter							
Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	t	t	t	t	none	f

Tables:

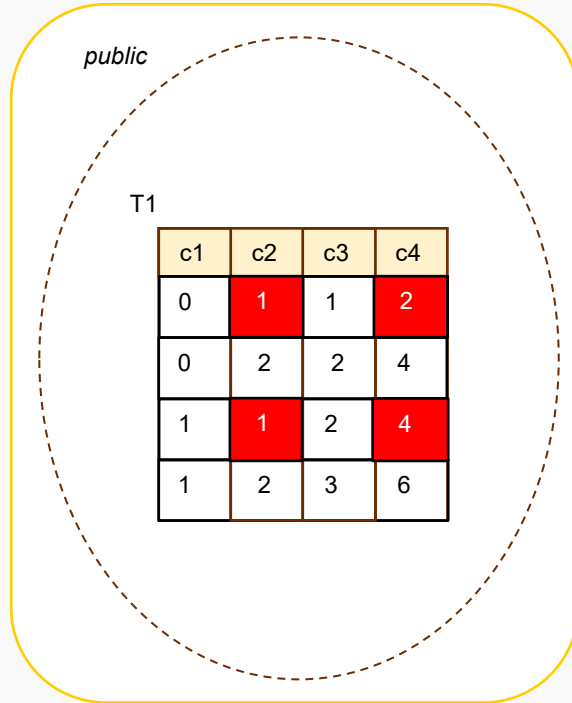
```
"public.t1" WHERE (c2 = 1)
```

```
test_pub=# SELECT * FROM pg_publication_tables;
```

pubname	schemaname	tablename	attnames	rowfilter
pub_rowfilter	public	t1	{c1,c2,c3,c4}	(c2 = 1)

(1 row)

Combo: Column List + Row Filter

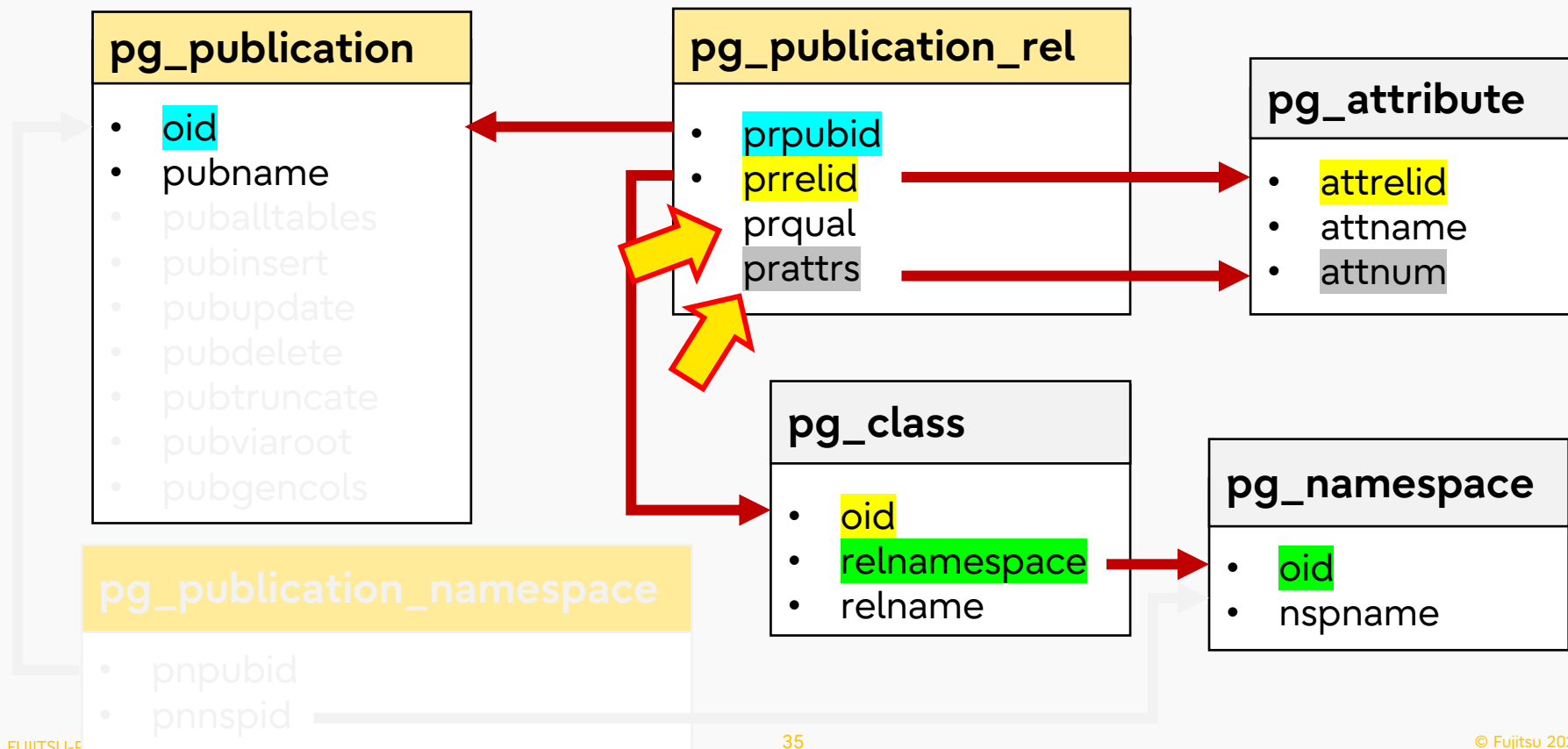


CREATE PUBLICATION **pub_combo**
FOR TABLE **t1(c2, c4) WHERE (c2 = 1);**

Publish:

- Only the specified tables
- Only data from the specified columns, from rows matching the specified condition

Combo: Column List + Row Filter



```
test_pub=# \dRp+
```

Publication pub_combo							
Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	t	t	t	t	none	f

Tables:

```
"public.t1" (c1, c4) WHERE (c1 = 1)
```

```
test_pub=# SELECT * FROM pg_publication_tables;
```

pubname	schemaname	tablename	attnames	rowfilter
pub_combo	public	t1	{c1,c4}	(c1 = 1)

(1 row)

Stored Generated Columns

```
CREATE TABLE tg1(c1 int, c2 int,  
                  gc1 int GENERATED ALWAYS AS (c1 + 50) STORED,  
                  gc2 int GENERATED ALWAYS AS (c2 * 100) STORED);
```

```
INSERT INTO tg1 VALUES (1,1), (2,2), (3,3);
```

```
SELECT * FROM tg1;
```

c1	c2	gc1	gc2
1	1	51	100
2	2	52	200
3	3	53	300

(3 rows)

Default
replication?

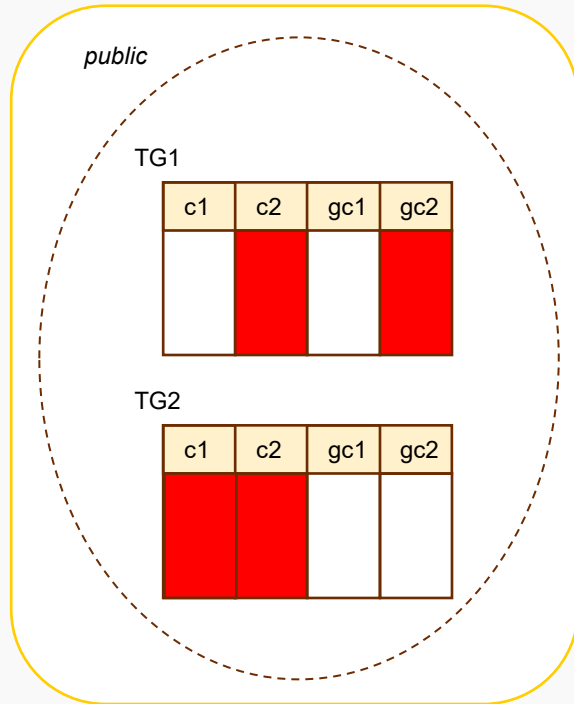
Don't replicate generated columns

Assumed subscriber column is generated

Generated
columns?

So, why
publish?

*For when subscriber column is **not** generated*



```
CREATE PUBLICATION pub_gencols  
FOR TABLE tg1(c2, gc2), tg2  
WITH (publish_generated_columns=none)
```

Since PG18

- If there is a **column list** (e.g. TG1) then publish all the specified columns regardless of the `publish_generated_columns` option. In this example, "TG1.gc2" will be published even though the option is false.
- If there is **no column list** (e.g. TG2), publication of the generated column data depends on the `publish_generated_columns` option. In this example TG2 does not publish generated columns because the option is `none`. If the option was `stored`, then all columns (c1, c2, gc1, gc2) of TG2 would have been published.

Publisher

Generated column



Subscriber

Generated column



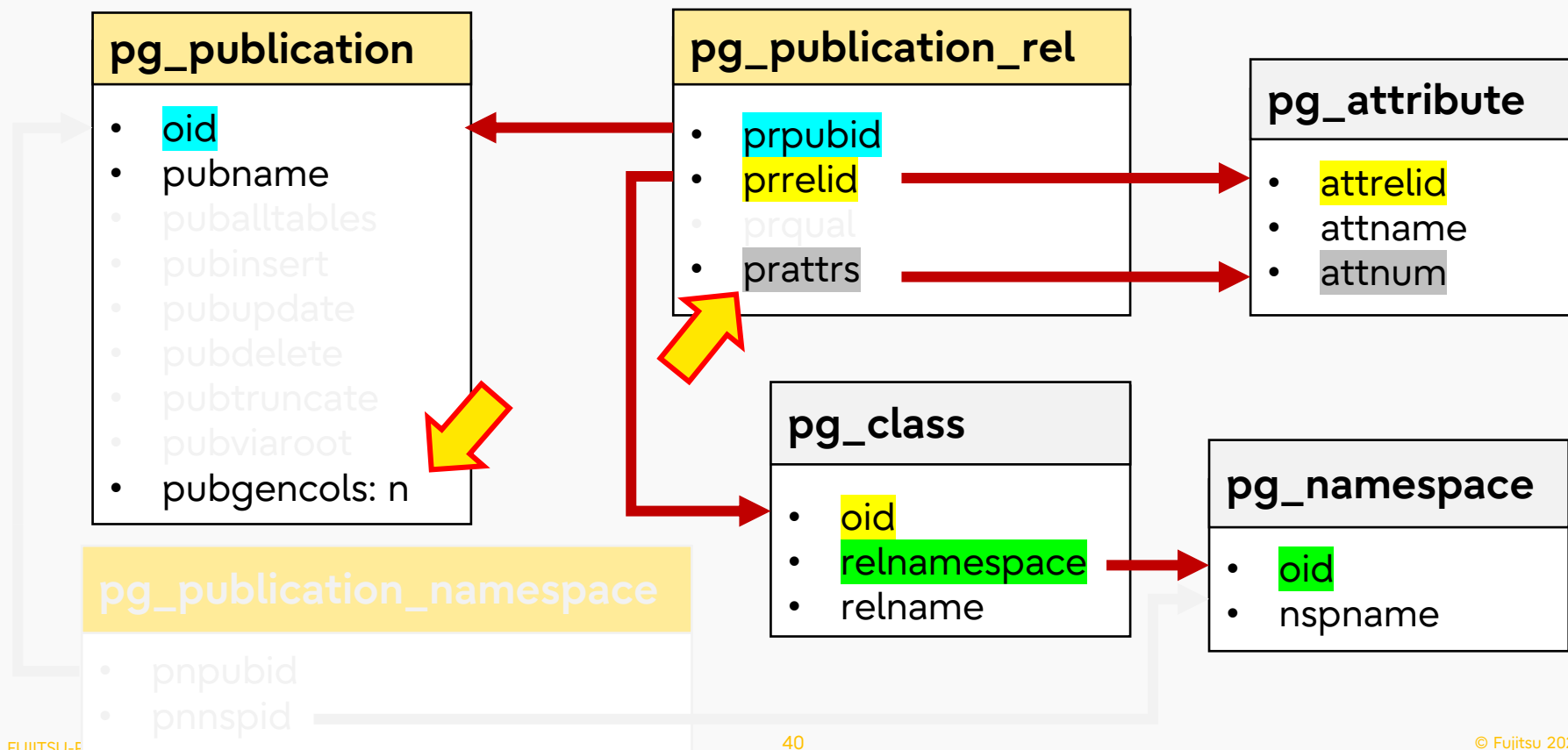
Generated column



Regular Column



Stored Generated Columns



```
test_pub=# \dRp+
```

Publication pub_gencols

Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	t	t	t	t	none	f

Tables:

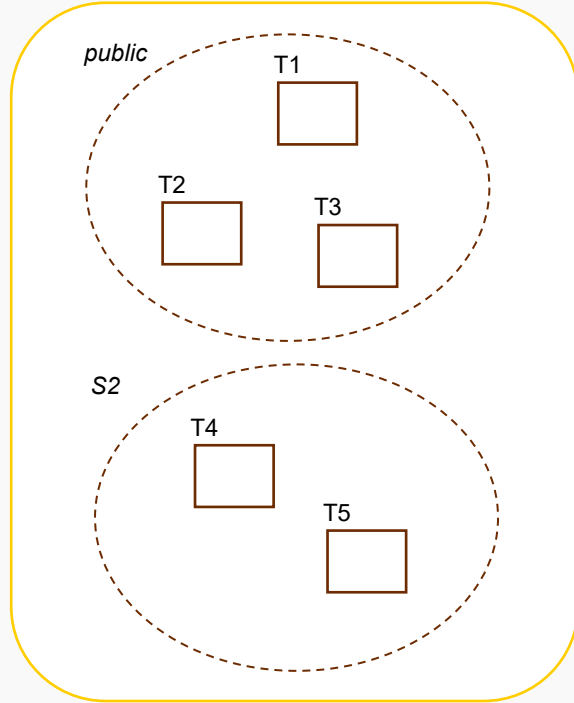
"public.tg1" (c2, gc2)

"public.tg2"

```
test_pub=# SELECT * FROM pg_publication_tables;
```

pubname	schemaname	tablename	attnames	rowfilter
pub_gencols	public	tg1	{c2,gc2}	
pub_gencols	public	tg2	{c1,c2}	

(2 rows)



CREATE PUBLICATION **pub_nothing**;

Since PG13

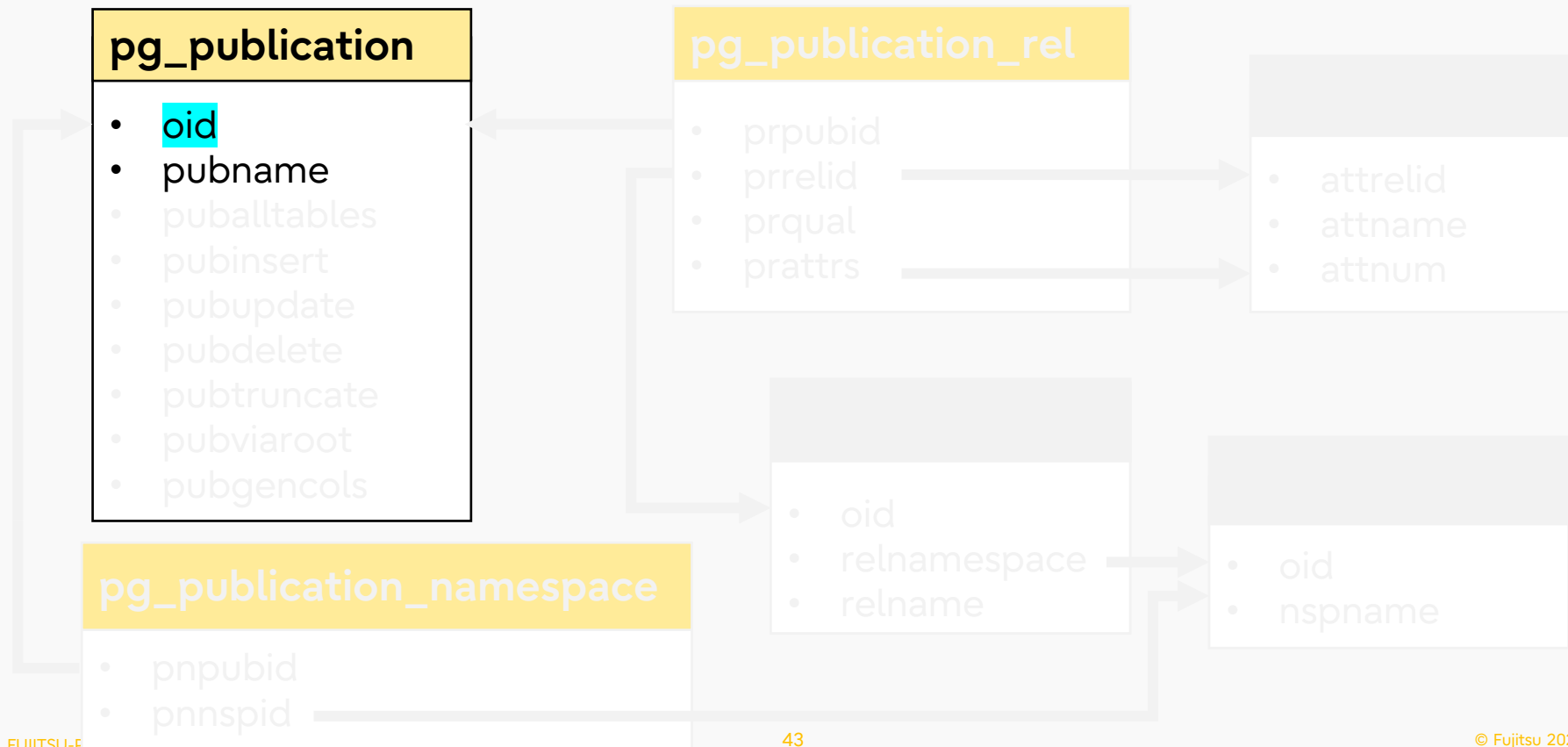
Publish:

- Nothing



Why?

"... is useful if tables or schemas are to be added later"

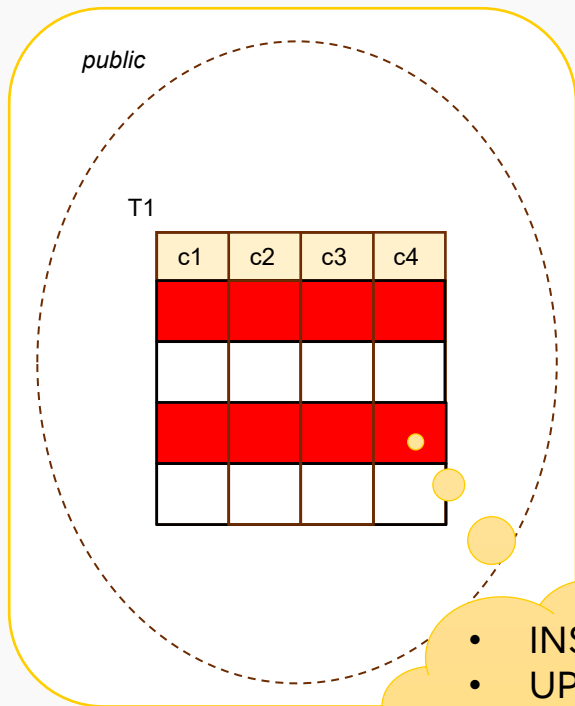


```
test_pub=# \dRp+
```

```
                                Publication pub_nothing
 Owner   | All tables | Inserts | Updates | Deletes | Truncates | Generated columns | Via root
-----+-----+-----+-----+-----+-----+-----+-----
 postgres | f          | t       | t       | t       | t       | none             | f
(1 row)
```

```
test_pub=# SELECT * FROM pg_publication_tables;
```

```
 pubname | schemaname | tablename | attnames | rowfilter
-----+-----+-----+-----+-----
(0 rows)
```



- INSERT?
- UPDATE?
- DELETE?
- TRUNCATE?

CREATE PUBLICATION `pub_dml_insertupdate`
FOR TABLE `t1` **WITH** (`publish='insert,update'`);

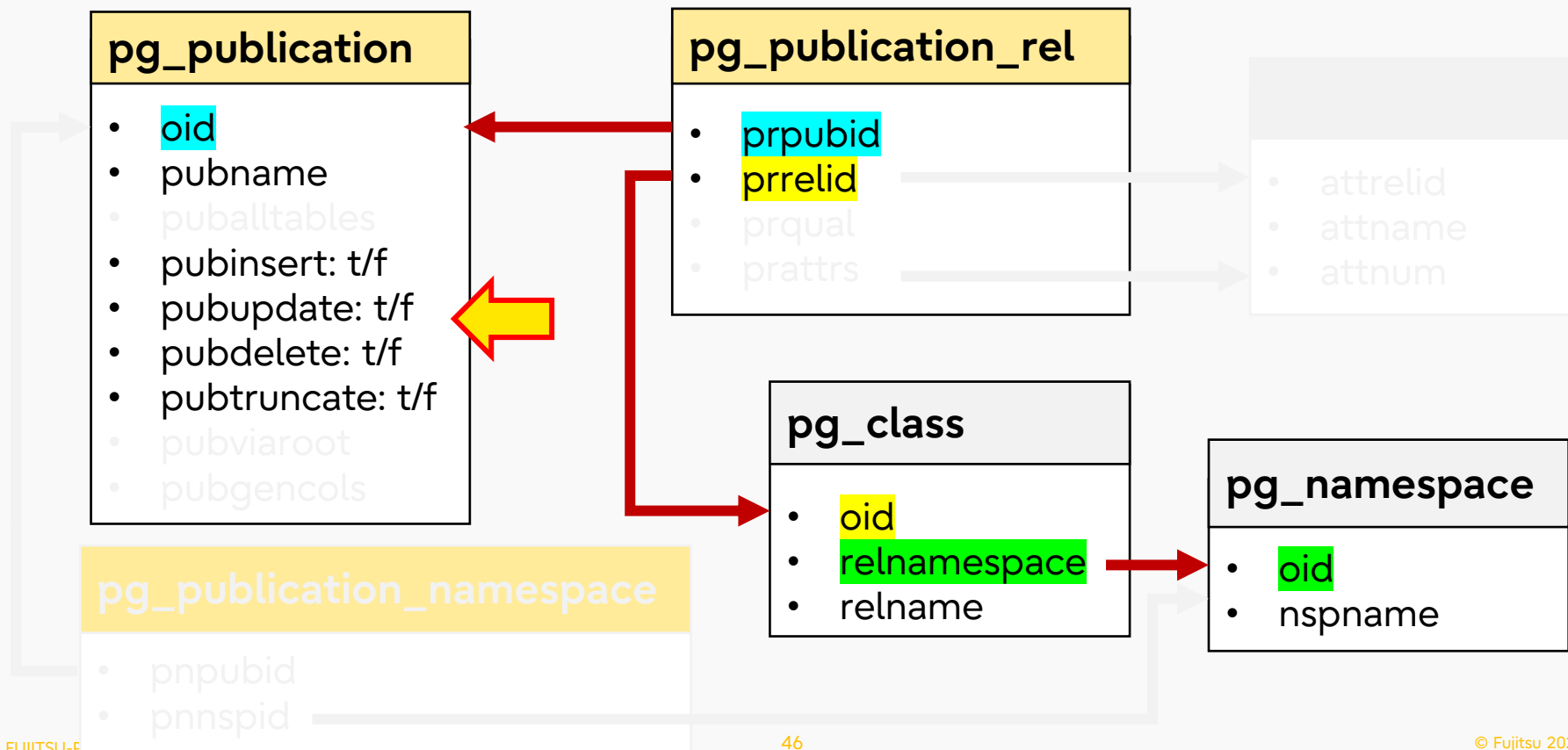
Since PG13

Publish:

- All schema, table, column data according to the CREATE PUBLICATION syntax
- Publish data only for permitted DML operations

Default?

Everything



```
CREATE PUBLICATION pub_dml_insertonly FOR TABLE t1 WITH (publish='insert');  
  
CREATE PUBLICATION pub_dml_insertupdate FOR TABLE t1 WITH (publish='insert,update');  
  
CREATE PUBLICATION pub_dml_noininsert FOR TABLE t1 WITH (publish='update,delete,truncate');  
  
CREATE PUBLICATION pub_dml_notruncate FOR TABLE t1 WITH (publish='insert,update,delete');  
  
CREATE PUBLICATION pub_dml_default FOR TABLE t1;  
  
CREATE PUBLICATION pub_dml_none FOR TABLE t1 WITH (publish='');
```

```
test_pub=# \dRp+
```

			Publication pub_dml_default					
Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root	
postgres	f	t	t	t	t	none	f	

Tables:
"public.t1"

Publication pub_dml_insertonly							
Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	t	f	f	f	none	f

Tables:

"public.t1"

Publication pub_dml_insertupdate							
Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	t	t	f	f	none	f

Tables:

"public.t1"

Publication pub_dml_noinstert							
Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	f	t	t	t	none	f

Tables:

"public.t1"

Publication pub_dml_none

Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	f	f	f	f	none	f

Tables:

"public.t1"

Publication pub_dml_notruncate

Owner	All tables	Inserts	Updates	Deletes	Truncates	Generated columns	Via root
postgres	f	t	t	t	f	none	f

Tables:

"public.t1"

```
test_pub=# SELECT oid, pubname, pubinsert, pubupdate, pubdelete, pubtruncate FROM pg_publication;
```

oid	pubname	pubinsert	pubupdate	pubdelete	pubtruncate
16401	pub_dml_insertonly	f	f	f	f
16403	pub_dml_insertupdate	t	t	f	f
16405	pub_dml_noininsert	f	t	t	t
16407	pub_dml_notruncate	t	t	t	f
16409	pub_dml_default	t	t	t	t
16411	pub_dml_none	f	f	f	f

(6 rows)

```
test_pub=# SELECT * FROM pg_publication_tables;
```

pubname	schemaname	tablename	attnames	rowfilter
pub_dml_insertonly	public	t1	{c1,c2,c3,c4}	
pub_dml_insertupdate	public	t1	{c1,c2,c3,c4}	
pub_dml_noininsert	public	t1	{c1,c2,c3,c4}	
pub_dml_notruncate	public	t1	{c1,c2,c3,c4}	
pub_dml_default	public	t1	{c1,c2,c3,c4}	
pub_dml_none	public	t1	{c1,c2,c3,c4}	

(6 rows)

Publisher

```
CREATE PUBLICATION pub_dml_insertonly  
FOR TABLE t1  
WITH (publish='insert');
```

```
INSERT INTO t1 VALUES (1,2,3,4);
```

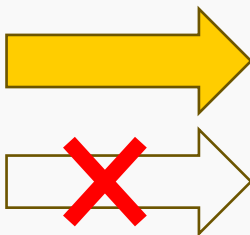
```
DELETE FROM t1 WHERE (c1=1);
```

```
SELECT * FROM t1;  
 c1 | c2 | c3 | c4  
----+----+----+----  
(0 rows)
```

Subscriber

```
CREATE SUBSCRIPTION sub  
CONNECTION 'dbname=test_pub'  
PUBLICATION pub_dml_insertonly;
```

```
SELECT * FROM t1;  
 c1 | c2 | c3 | c4  
----+----+----+----  
  1 |  2 |  3 |  4  
(1 row)
```

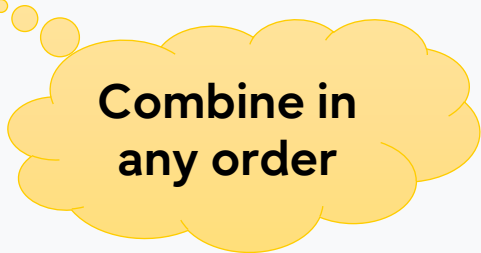


Combos and Flexibility

More Combinations

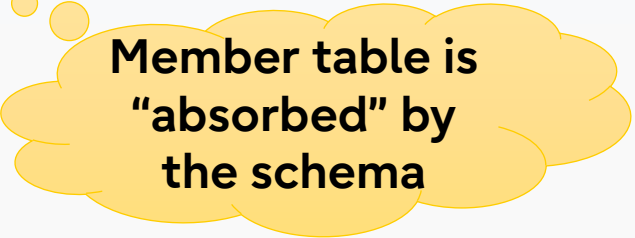
	ALL TABLES	TABLES IN SCHEMA <i>s</i>	TABLE <i>t</i>	TABLE <i>t</i> (column list)	WHERE (row filter)	WITH (gen cols)	WITH (dml ops)
ALL TABLES						✓	✓
TABLES IN SCHEMA <i>s</i>		✓	✓	✓		✓	✓
TABLE <i>t</i>		✓	✓	✓	✓	✓	✓
TABLE <i>t</i> (column list)		✓	✓	✓	✓	✓	✓
WHERE (row filter)			✓	✓	✓	✓	✓
WITH (publish generated cols)	✓	✓	✓	✓	✓		✓
WITH (publish dml operations)	✓	✓	✓	✓	✓	✓	

- `CREATE PUBLICATION pub FOR TABLES IN SCHEMA s2, TABLE t1;`
`CREATE PUBLICATION pub FOR TABLE t1, TABLES IN SCHEMA s2;`



**Combine in
any order**

- `CREATE PUBLICATION pub FOR TABLES IN SCHEMA s2, TABLE s2.t4;`



**Member table is
"absorbed" by
the schema**

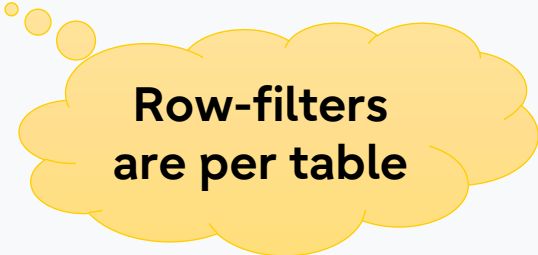
- CREATE PUBLICATION pub FOR TABLE t1, t2, t3;
CREATE PUBLICATION pub FOR TABLE t1, TABLE t2, t3;
CREATE PUBLICATION pub FOR TABLE t1, TABLE t2, TABLE t3;



**Short &
Long**

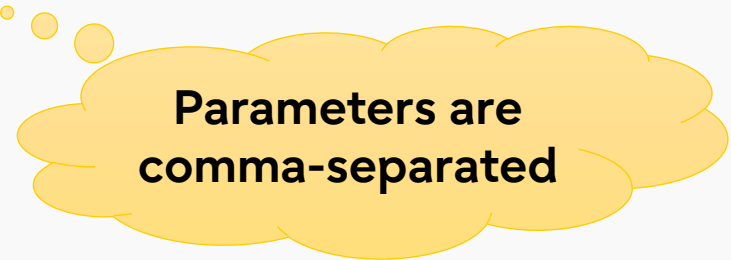
- CREATE PUBLICATION pub FOR TABLES IN SCHEMA s1, s2, s3;
CREATE PUBLICATION pub FOR TABLES IN SCHEMA s1, TABLES IN
SCHEMA s2, TABLES IN SCHEMA s3;

- `CREATE PUBLICATION pub FOR TABLE t1 WHERE (c1>2), t2 WHERE (c3=6) ;`



**Row-filters
are per table**

- `CREATE PUBLICATION pub FOR TABLE t1 WITH (publish='insert,delete', publish_generated_columns=stored) ;`



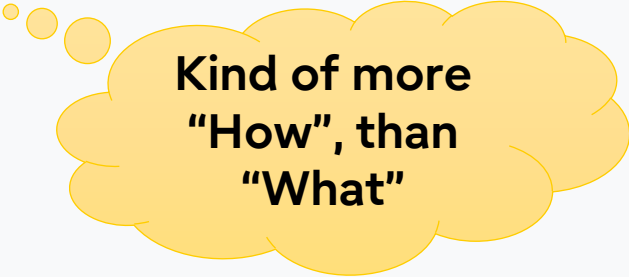
**Parameters are
comma-separated**

Things I did not talk about

- **CREATE SUBSCRIPTION ...**
- **ALTER PUBLICATION ...**
- **ALTER SUBSCRIPTION REFRESH PUBLICATION
versus "Future" tables**
- **DDL replication**
- **REPLICA IDENTITY**

- For Partition tables:

`publish_via_partition_root` (t/f) option



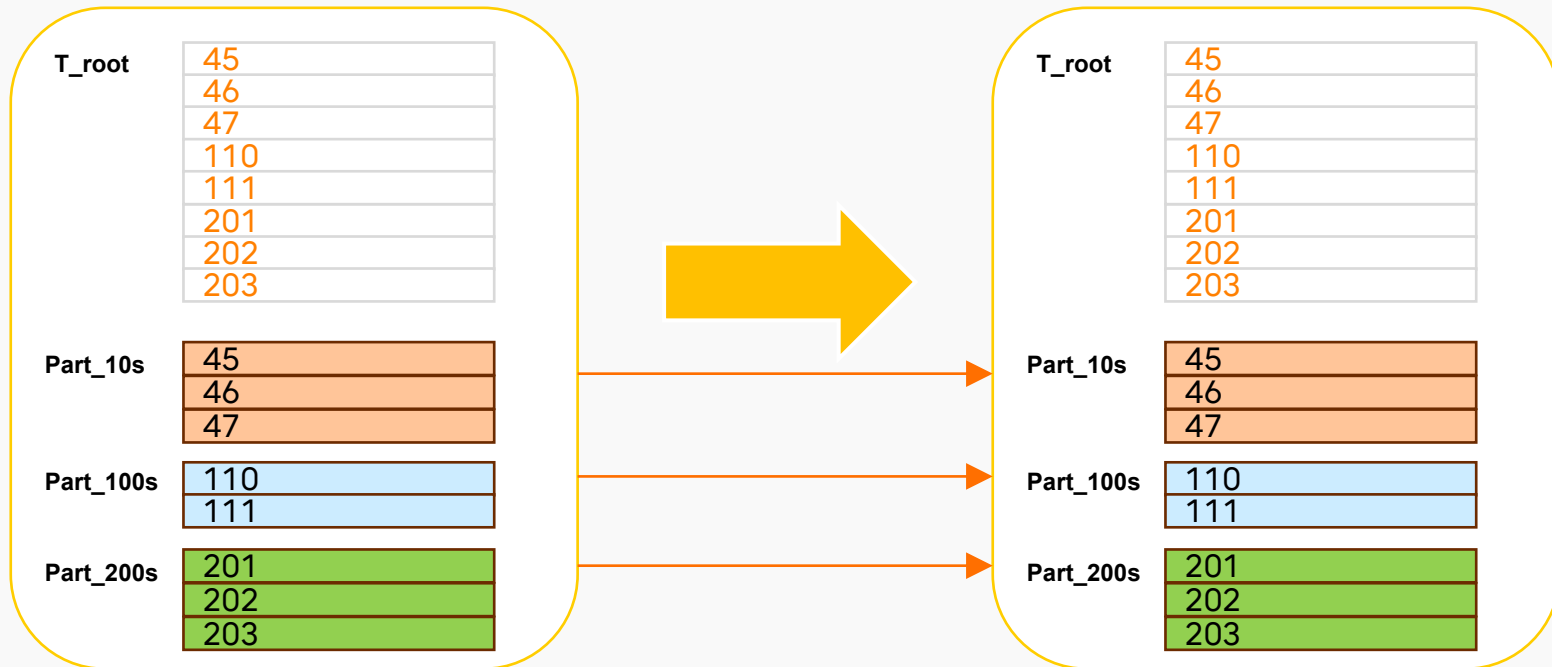
Kind of more
“How”, than
“What”

- For Descendant tables:

`ONLY` and `*`

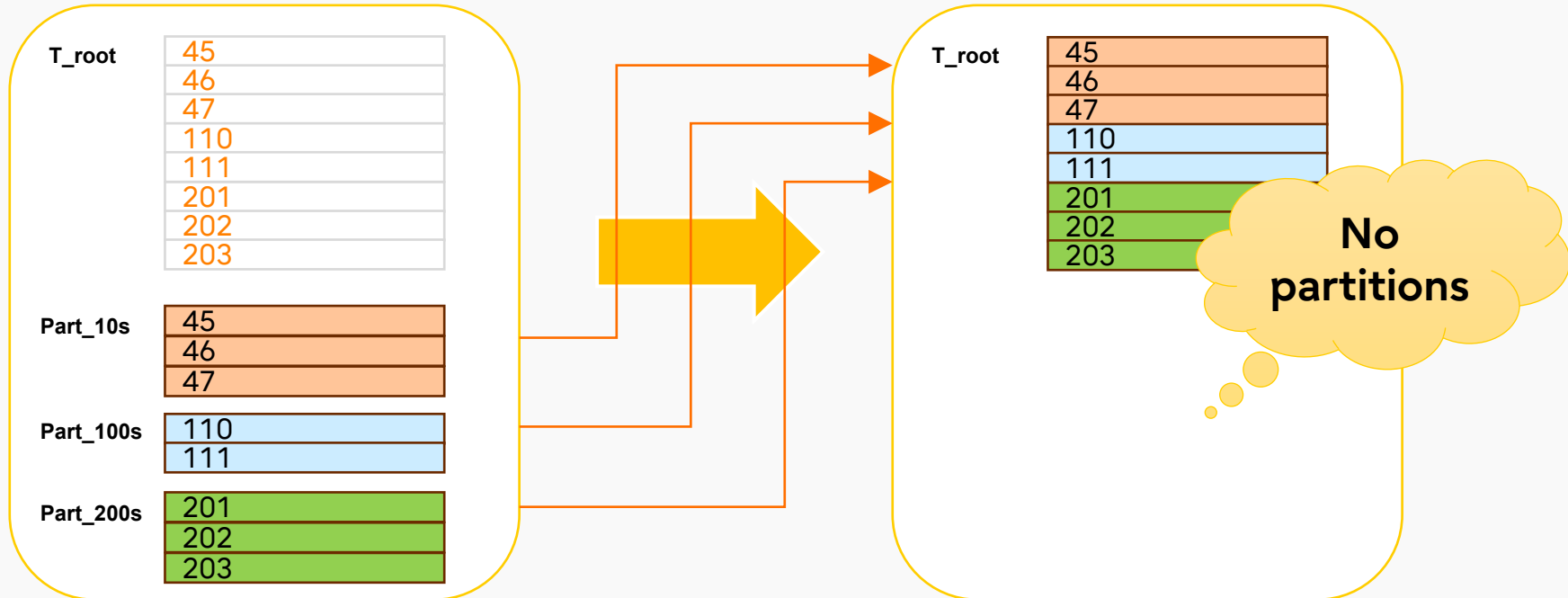
Partition Tables

```
CREATE PUBLICATION mypub ...  
WITH (publish_via_partition_root=false)
```



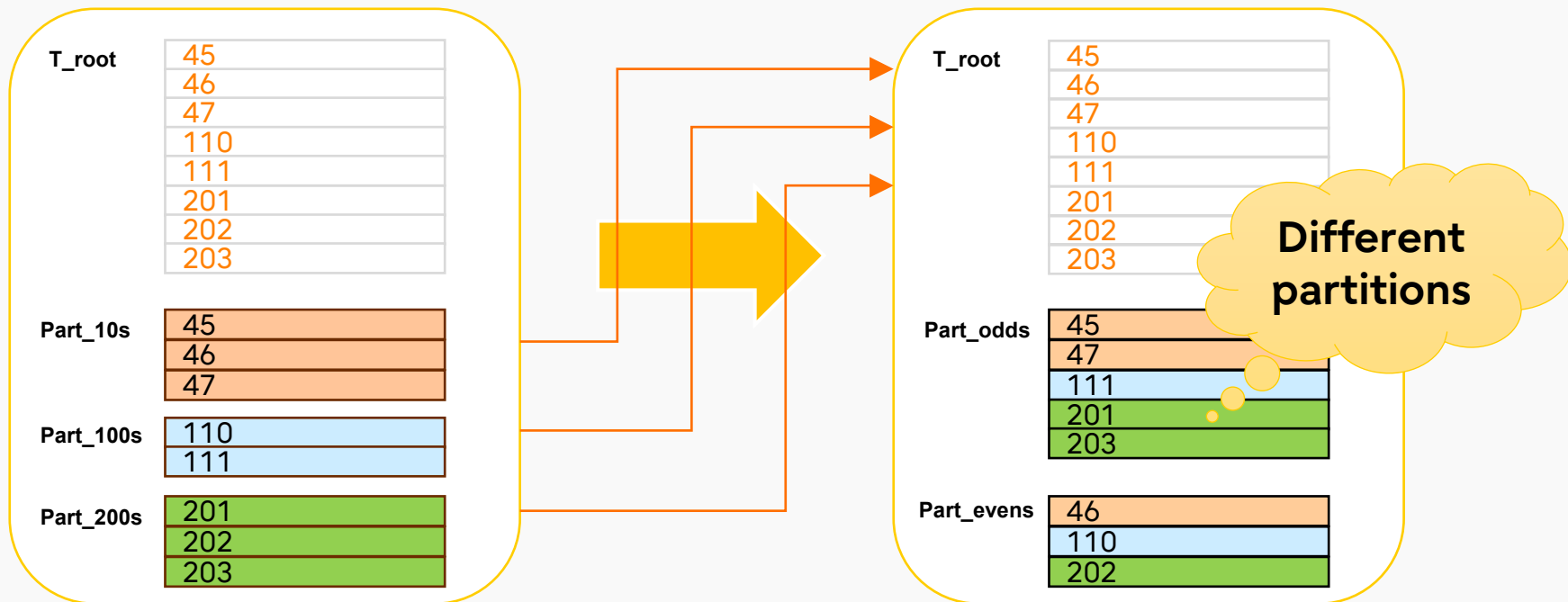
Partition Tables

```
CREATE PUBLICATION mypub ...  
WITH (publish_via_partition_root=true)
```



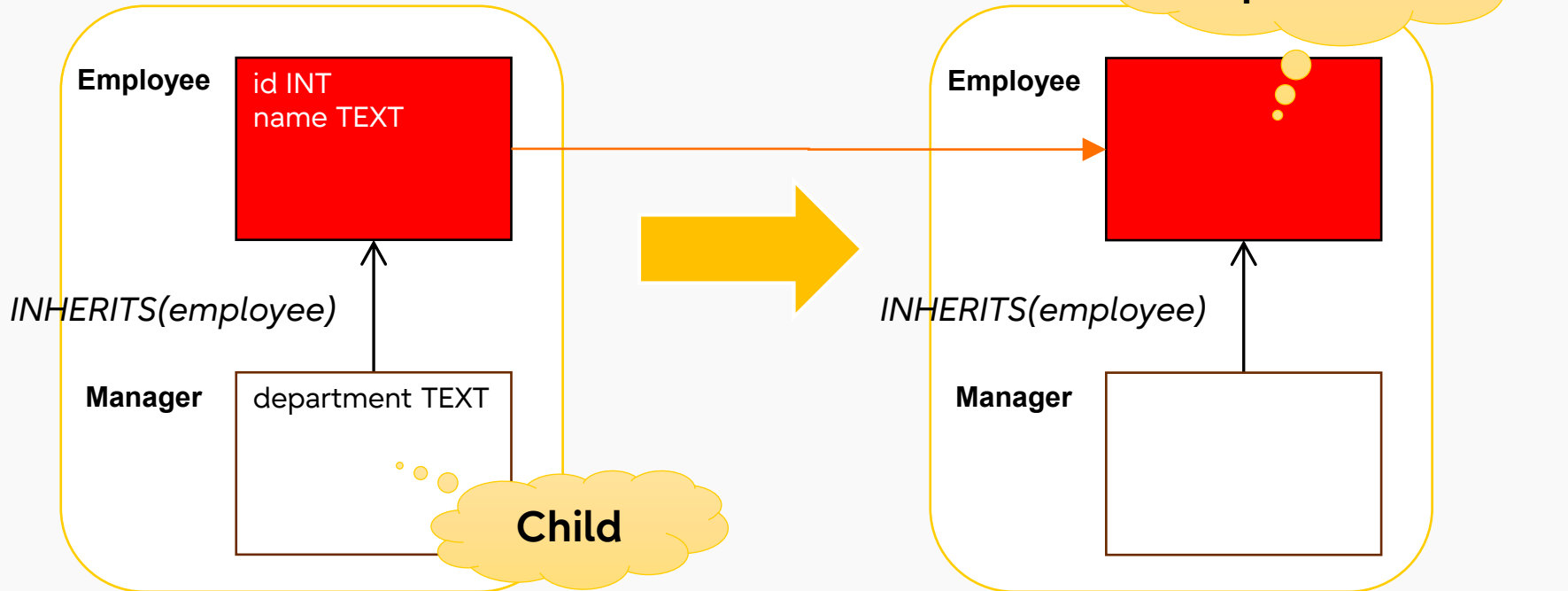
Partition Tables

```
CREATE PUBLICATION mypub ...  
WITH (publish_via_partition_root=true)
```



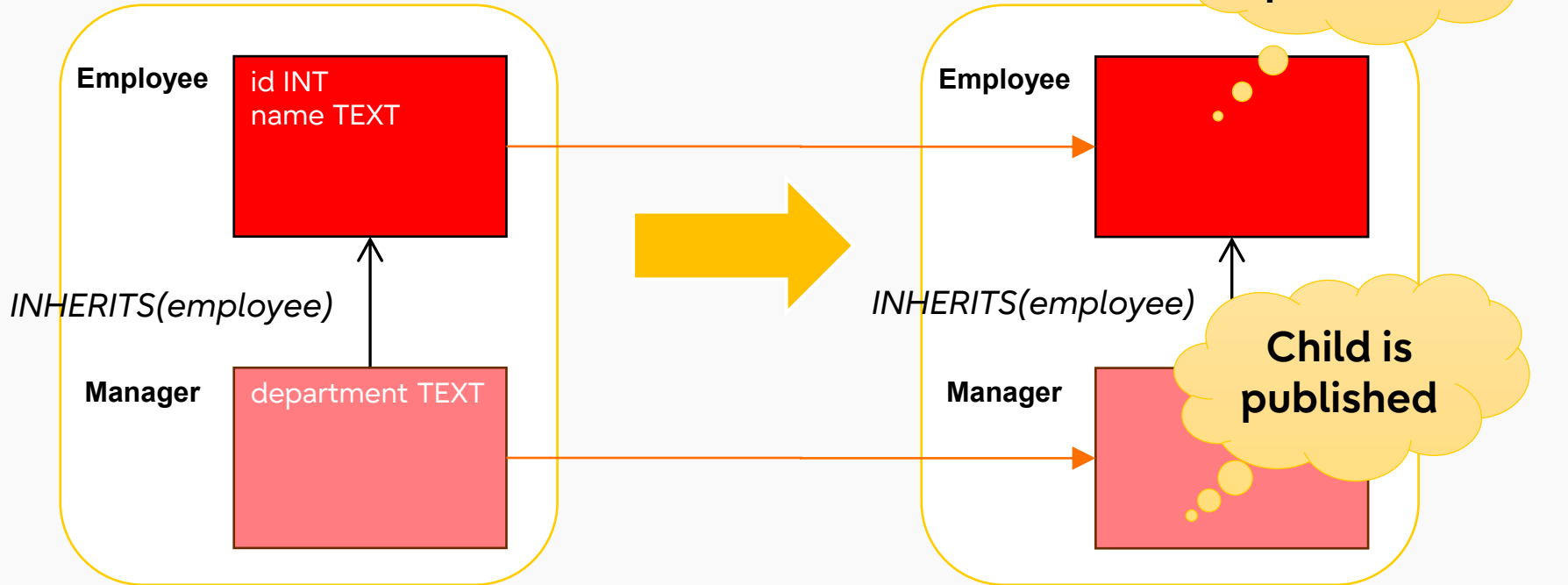
Descendant Tables

CREATE PUBLICATION *mypub*
FOR TABLE **ONLY** *employee*



Descendant Tables

```
CREATE PUBLICATION mypub  
FOR TABLE employee *
```



Future Knobs (PG19)

- ... FOR ALL SEQUENCES ;
- ... FOR SEQUENCES IN SCHEMA *s1* ;
- ... FOR SEQUENCE *seq1*, *seq2* ;

tables:

- ... FOR ALL TABLES **EXCEPT t1,t2**;
- ... FOR TABLES IN SCHEMA s1 **EXCEPT t1,t2**;



Why?

Easier

"Future" tables

columns:

- ... FOR TABLE t1 **EXCEPT (c1,c2)**;

Summary

```
CREATE PUBLICATION name
[ FOR ALL TABLES |
  FOR publication_object [, ...] ]
[ WITH ( publication_parameter [= value] [, ...] ) ]
```

where *publication_object* is one of:

```
TABLE { [ ONLY ] table_name [ * ] [ ( column_name [, ...] ) ]
      [ WHERE ( expression ) ] } [, ...]
TABLES IN SCHEMA { schema_name | CURRENT_SCHEMA } [, ...]
```

CREATE PUBLICATION *name*

[**FOR ALL TABLES** |

FOR *publication_object* [, ...]]

[**WITH** (*publication_parameter* [= *value*] [, ...])]



All tables

where *publication_object* is one of:

TABLE { [**ONLY**] *table_name* [*****] [(*column_name* [, ...])]

[**WHERE** (*expression*)] } [, ...]

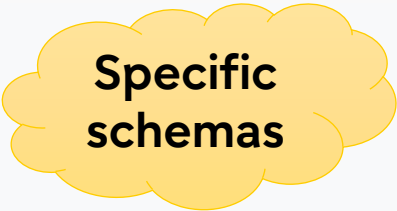
TABLES IN SCHEMA { *schema_name* | **CURRENT_SCHEMA** } [, ...]

```
CREATE PUBLICATION name
```

```
[ FOR ALL TABLES |
```

```
FOR publication_object [, ...] ]
```

```
[ WITH ( publication_parameter [= value] [, ...] ) ]
```



Specific
schemas

where *publication_object* is one of:

```
TABLE { [ ONLY ] table_name [ * ] [ ( column_name [, ...] ) ]
```

```
[ WHERE ( expression ) ] } [, ...]
```

```
TABLES IN SCHEMA { schema_name | CURRENT_SCHEMA } [, ...]
```

```
CREATE PUBLICATION name
```

```
[ FOR ALL TABLES |
```

```
FOR publication_object [, ...] ]
```

```
[ WITH ( publication_parameter [= value] [, ...] ) ]
```



Specific
tables

where *publication_object* is one of:

```
TABLE { [ ONLY ] table_name [ * ] [ ( column_name [, ...] ) ]
```

```
[ WHERE ( expression ) ] } [, ...]
```

```
TABLES IN SCHEMA { schema_name | CURRENT_SCHEMA } [, ...]
```

CREATE PUBLICATION *name*

[**FOR ALL TABLES** |

FOR *publication_object* [, ...]]

[**WITH** (*publication_parameter* [= *value*] [, ...])]



Column
lists

where *publication_object* is one of:

TABLE { [**ONLY**] *table_name* [*****] [(*column_name* [, ...])]

[**WHERE** (*expression*)] } [, ...]

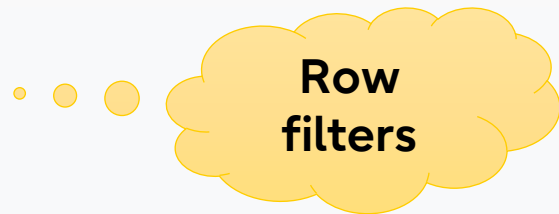
TABLES IN SCHEMA { *schema_name* | **CURRENT_SCHEMA** } [, ...]

CREATE PUBLICATION *name*

[FOR ALL TABLES |

FOR *publication_object* [, ...]]

[WITH (*publication_parameter* [= *value*] [, ...])]



where *publication_object* is one of:

TABLE { [ONLY] *table_name* [*] [(*column_name* [, ...])]

[**WHERE** (*expression*)] } [, ...]

TABLES IN SCHEMA { *schema_name* | CURRENT_SCHEMA } [, ...]

CREATE PUBLICATION *name*

[FOR ALL TABLES |

FOR *publication_object* [, ...]]

[**WITH** (*publication_parameter* [= *value*] [, ...])]



where *publication_object* is one of:

TABLE { [**ONLY**] *table_name* [*****] [(*column_name* [, ...])]

[**WHERE** (*expression*)] } [, ...]

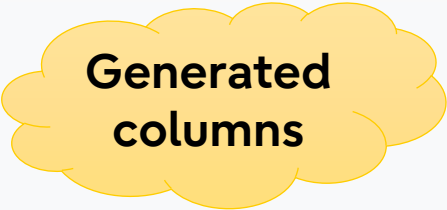
TABLES IN SCHEMA { *schema_name* | **CURRENT_SCHEMA** } [, ...]

CREATE PUBLICATION *name*

[**FOR ALL TABLES** |

FOR *publication_object* [, ...]]

[**WITH** (*publication_parameter* [= *value*] [, ...])]



Generated
columns

where *publication_object* is one of:

TABLE { [**ONLY**] *table_name* [*****] [(*column_name* [, ...])]

[**WHERE** (*expression*)] } [, ...]

TABLES IN SCHEMA { *schema_name* | **CURRENT_SCHEMA** } [, ...]

```
CREATE PUBLICATION name
```

```
[ FOR ALL TABLES |
```

```
  FOR publication_object [, ...] ]
```

```
[ WITH ( publication_parameter [= value] [, ...] ) ]
```



where *publication_object* is one of:

```
TABLE { [ ONLY ] table_name [ * ] [ ( column_name [, ...] ) ]
```

```
  [ WHERE ( expression ) ] } [, ...]
```

```
TABLES IN SCHEMA { schema_name | CURRENT_SCHEMA } [, ...]
```

CREATE PUBLICATION *name*

[**FOR ALL TABLES** |

FOR *publication_object* [, ...]]

[**WITH** (*publication_parameter* [= *value*] [, ...])]



DML
operations

where *publication_object* is one of:

TABLE { [**ONLY**] *table_name* [*****] [(*column_name* [, ...])]

[**WHERE** (*expression*)] } [, ...]

TABLES IN SCHEMA { *schema_name* | **CURRENT_SCHEMA** } [, ...]

CREATE PUBLICATION *name*

[**FOR ALL TABLES** |

FOR *publication_object* [, ...]]

[**WITH** (*publication_parameter* [= *value*] [, ...])]



where *publication_object* is one of:

TABLE { [**ONLY**] *table_name* [*****] [(*column_name* [, ...])]

[**WHERE** (*expression*)] } [, ...]

TABLES IN SCHEMA { *schema_name* | **CURRENT_SCHEMA** } [, ...]

CREATE PUBLICATION *name*

[**FOR ALL TABLES** |

FOR *publication_object* [, ...]]

[**WITH** (*publication_parameter* [= *value*] [, ...])]



where *publication_object* is one of:

TABLE { [**ONLY**] *table_name* [*****] [(*column_name* [, ...])]

[**WHERE** (*expression*)] } [, ...]

TABLES IN SCHEMA { *schema_name* | **CURRENT_SCHEMA** } [, ...]

CREATE PUBLICATION *name*

[FOR ALL TABLES |

FOR *publication_object* [, ...]]

[WITH (*publication_parameter* [= *value*] [, ...])]



Descendant
tables

where *publication_object* is one of:

TABLE { [**ONLY**] *table_name* [*] [(*column_name* [, ...])]

[**WHERE** (*expression*)] } [, ...]

TABLES IN SCHEMA { *schema_name* | CURRENT_SCHEMA } [, ...]

```
CREATE PUBLICATION name
```

```
[ FOR ALL TABLES |
```

```
FOR publication_object [, ...] ]
```

```
[ WITH ( publication_parameter [= value] [, ...] ) ]
```



Descendant
tables

where *publication_object* is one of:

```
TABLE { [ ONLY ] table_name [ * ] [ ( column_name [, ...] ) ]
```

```
[ WHERE ( expression ) ] } [, ...]
```

```
TABLES IN SCHEMA { schema_name | CURRENT_SCHEMA } [, ...]
```

References

- **PostgreSQL Documentation**

- **CREATE PUBLICATION**

- <https://www.postgresql.org/docs/18/sql-createpublication.html>

- **Logical Replication Examples**

- <https://www.postgresql.org/docs/18/logical-replication.html>

- **Blog**

- **Generated Column replication**

- <https://www.postgresql.fastware.com/pzone/2025-04-logical-replication-of-generated-columns>

- **PG19 pgsql-hackers threads (WIP)**

- **SEQUENCES** – [“Logical Replication of sequences”](#)

- **EXCEPT** – [“Skipping schema changes in publication”](#)

Thank you



Q & A