



UPDATE
SET <column>=value

4	song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date date
1	2	SQL song	Not defined		2022-01-07
2	3	SQL song2	Not defined	0.99	2022-01-07
3	4	SQL song3	Not defined	0.99	2022-01-07

UPDATE songs
SET genre='Country music'

4	song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date date
1	2	SQL song	Not defined		2022-01-07
2	3	SQL song2	Not defined	0.99	2022-01-07
3	4	SQL song3	Not defined	0.99	2022-01-07

UPDATE songs
SET genre='Country music'

song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date date
2	SQL song	Country music	0.99	2022-01-07
3	SQL song2	Country music	0.99	2022-01-07
4	SQL song3	Country music	0.99	2022-01-07

UPDATE songs SET genre='Pop music' WHERE song_id=4

song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date date
2	SQL song	Country music	0.99	2022-01-07
3	SQL song2	Country music	0.99	2022-01-07
4	SQL song3	Country music	0.99	2022-01-07

UPDATE songs SET genre='Pop music' WHERE song_id=4

song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date. date
2	SQL song	Country music	0.99	2022-01-07
3	SQL song2	Country music	0.99	2022-01-07
4	SQL song3	Pop music	0.99	2022-01-07

UPDATE songs
SET price=song_id+0.99

song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date.date
2	SQL song	Country music	0.99	2022-01-07
3	SQL song2	Country music	0.99	2022-01-07
4	SQL song3	Pop music	0.99	2022-01-07

UPDATE songs
SET price=song_id+0.99

song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date date
2	SQL song	Country music	2.99	2022-01-07
3	SQL song2	Country music	3.99	2022-01-07
4	SQL song3	Pop music	4.99	2022-01-07

INSERT





INSERT

INSERT INTO online_sales
(customer_id, film_id,amount)
VALUES (269,13,10.99)

			•		
4	transaction_id. [PK] integer	customer_id integer	film_id integer	amount numeric (5,2)	promotion character varying (10)
1	1	269	13	10.99	None

SERIAL

DEFAULT

Challenge

Update all rental prices that are 0.99 to 1.99.

The customer table needs to be altered as well:

- 1. Add the column *initials* (data type varchar(10))
- 2. Update the values to the actual initials for example Frank Smith should be F.S.

Challenge

Create a table called *users* with the following columns:



- 1. During creation add the DEFAULT current_date to the signup_date.
- 2. Add the constraint *namelength* to ensure the user_name has more than 2 characters.
- 3. Add the constraint with default name to ensure the birthdate is after 01-01-1900.
- 4. After the creation rename namelength to name_length.
- 5. Try to add Frank Smith with user name franksmith1 and birthday 02-12-1905.
- 6. Modify the constraint on the birthdate so that no dates after 01-01-1910 are allowed.
- 7. Try again to add Frank Smith with user name franksmith1 and birthday 02-12-1905.

Challenge

During creation add the DEFAULT 'Not defined' to the genre.

- 2. Add the not null constraint to the song_name column
- 3. Add the constraint with default name to ensure the price is at least 1.99.
- 4. Add the constraint *date_check* to ensure the release date is between today and 01-01-1950.
- 5. Try to add Frank Smith with user name franksmith1 and birthday 02-12-1905.
- 6. Modify the constraint on the birthdate so that no dates after 01-01-1910 are allowed.
- 7. Try again to add Frank Smith with user name franksmith1 and birthday 02-12-1905.

Constraints

COLUMN CONSTRAINTS

What constraints do we have?

NOT NULL

UNIQUE

DEFAULT

Ensures that a column cannot have a NULL value

Ensures that all values in a column are different

Sets a default value for a column if no value is specified

ERROR: insert or update on table "director" violates foreign key constraint "director_address_id_fkey"

DETAIL: Key (address_id)=(0) is not present in table "address".

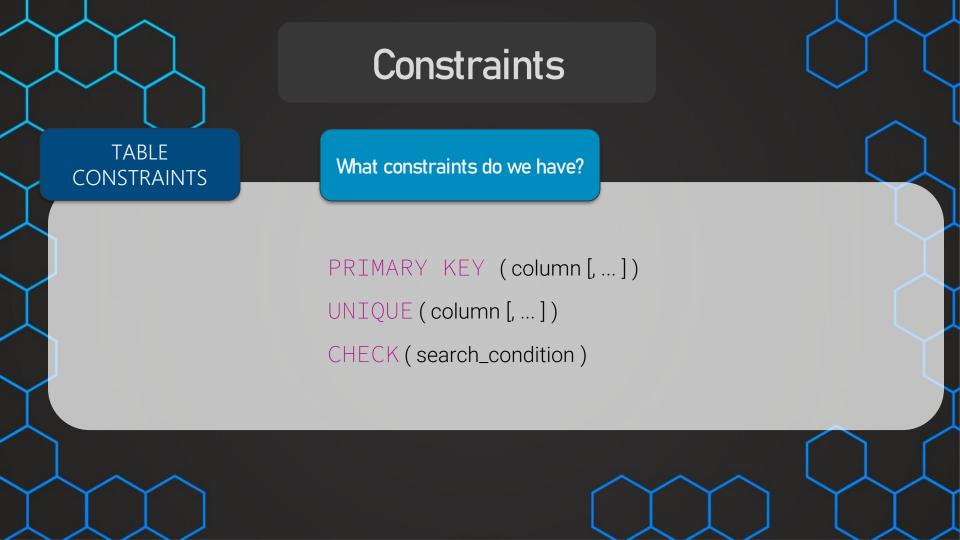
SQL state: 23503

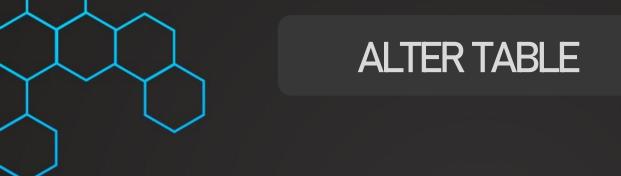
REFERENCES

CHECK

Ensures referential integrity (only values of another column can be used)

Ensures that the values in a column satisfies a specific condition





ALTER TABLE <table_name>
ALTER COLUMN <column_name> SET DEFAULT <value>

DROP ADD TYPE RENAME DEFAULT

ALTER TABLE

ALTER TABLE staff
RENAME COLUMN first_name TO name,
DROP COLUMN last_name

CONSTRAINT

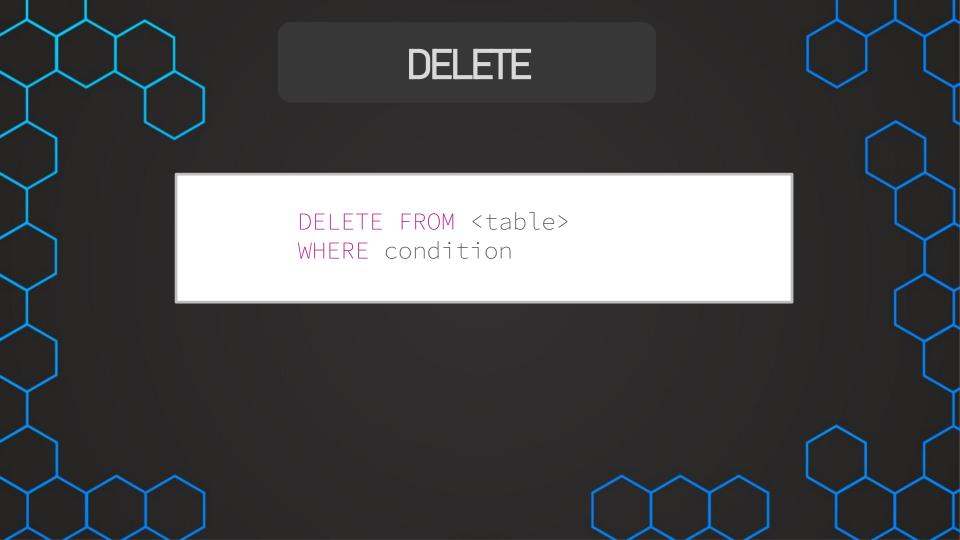
ALTER TABLE

ALTER TABLE <table_name>
ALTER COLUMN <column_name>
DROP DEFAULT <value>

DROP ADD ALTER TYPE RENAME DEFAULT

CONSTRAINT





DELETE FROM songs WHERE song_id=4

4	song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date date
1	2	SQL song	Not defined	0.99	2022-01-07
2	3	SQL song2	Not defined	0.99	2022-01-07
_		001	N I C. I	0.00	0000 04 07
J	4	SQL SUNGS	Not defined	0.99	2022-01-07

DELETE FROM songs
WHERE song_id IN (3,4)

4	song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date.date
1	2	SQL song	Not defined	0.99	2022-01-07
0	2	0010	Not defined	0.00	0000 01 07
	9	OQL SUNGE	Not defined	0.77	2022-01-07
		001	N . I C . I	0.00	0000 04 07
٥	4	SQL SUNGS	Not defined	0.55	2022-01-07

DELETE FROM songs

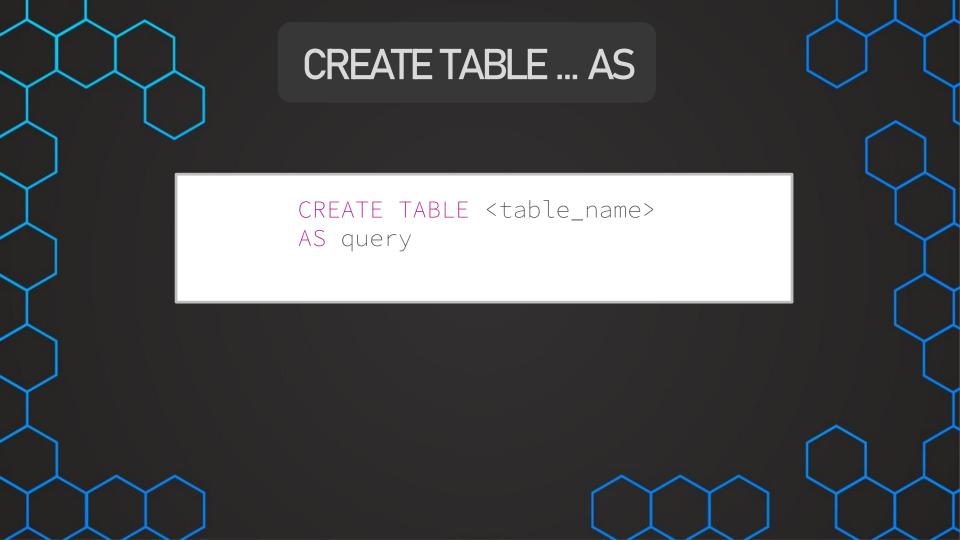
4	song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date date
1	2	COL cong	Not defined	0.00	2022 01 07
	_	od Looning	Hot defined	0.77	2022 01 07
0	0	001	Not defined	0.00	0000 01 07
_	0	OQL SUNGZ	Not defined	0.55	2022-01-07
_		001	N . I C . I	0.00	0000 04 07
3	4	SQL Sorigs	Not defined	0.99	2022-01-07

DELETE FROM songs
WHERE song_id IN (3,4)
RETURNING song_id



DELETE FROM songs WHERE song_id IN (3,4) RETURNING *

song_id song_name [PK] integer character varying (30)		genre character varying (30)	price numeric (4,2)	release_date date
3	SQL song2	Country music	3.99	2022-01-07
4	SQL song3	Pop music	4.99	2022-01-07





CREATE TABLE ... AS

CREATE TABLE customer_anonymous
AS
SELECT customer_id, initials
FROM customer
WHERE first_name LIKE 'C%'

CREATE TABLE ... AS

CREATE TABLE customer_anonymous

AS

SELECT customer_id, initials

FROM customer

WHERE first_name LIKE 'C%'

SELECT * FROM customer_anonymous

Pyhsical storage needed!

Data can change!

Alternative: Create a view and just store the statement!



CREATE VIEW customer_anonymous
AS
SELECT customer_id, initials
FROM customer
WHERE first_name LIKE 'C%'

CREATE VIEW customer_anonymous
AS
SELECT customer_id, initials
FROM customer
WHERE first_name LIKE 'C%'

SELECT * FROM customer_anonymous

If the query is slow the view will be slow!

CREATE TABLE customer_an_table
AS
SELECT * FROM customer_anonymous

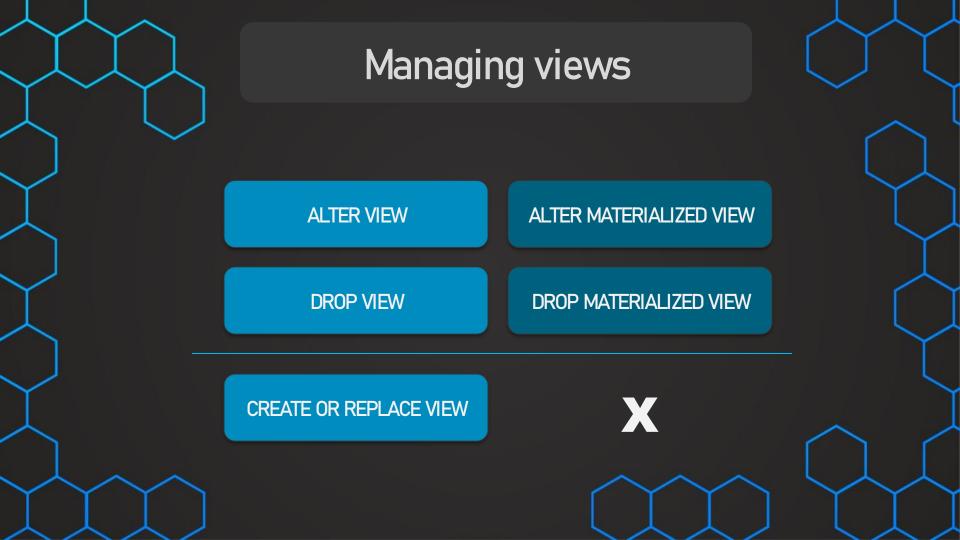
Problem: That table will not be updated if data in the underlying tables change!



Data Output Explain Messages Notifications

4	customer_id_integer	name text	address text	postal_code text	phone text	city text	country text
1	1	MARY BROWN	1913 Hanoi Way	35200	28303384290	Sasebo	Japan
2	2	PATRICIA JOHNSON	1121 Loja Avenue	17886	838635286649	San Bernardino	United States
3	3	LINDA WILLIAMS	692 Joliet Street	83579	448477190408	Athenai	Greece

Problem: That table will not be updated if data in the underlying tables change!





ALTER VIEW

ALTER VIEW customer_anonymous RENAME TO v_customer_info

ALTER VIEW v_customer_info
RENAME COLUMN name TO customer_name



If the query is slow the view will be slow!

CREATE TABLE customer_an_table
AS
SELECT * FROM customer_anonymous

Problem: That table will not be updated if data in the underlying tables change!

VIEW

UPDATE songs
SET genre='Country music'

	song_id [PK] integer	song_name character varying (30)	genre character varying (30)	price numeric (4,2)	release_date date
1	2	SQL song	Not defined	0.99	2022-01-07
2	3	SQL song2	Not defined	0.99	2022-01-07
3	4	SQL song3	Not defined	0.99	2022-01-07

Import & Export

Import external data into an existing table

Table needs to be created first!

Data needs to be in correct format!

Export data from a table into a csy file