Managing tables

CREATE

ALTER

DROP

Database

Table

Schemas

Managing tables

CREATE

ALTER

DROP

Data Definition

INSERT

UPDATE

DELETE

Data Manipulation

Data Structures

Data itself

Managing tables

CREATE

INSERT

Data Types

ALTER

Data Definition

UPDATE

Data Manipulation

Constraints

DROP

DELETE

Primary & Foreign Keys

Data Structures

Data itself

Views

Creating database

Very simple

```
CREATE DATABASE <database_name>;
```

Dropping database

Very simple

```
DROP DATABASE <database_name>;
```

Be very careful with dropping database objects!

Data Types

Important when creating tables

Understanding data types

Storage size

Differences

Allowed values

When to use which one?

Possible operations

How to store ZIP codes?

Data Types

Numeric

Strings

Date/Time

Other

https://www.postgresql.org/docs/current/datatype.html

Data Types

Numeric

Type	Storage size	Range	Notes
INT	4 bytes	-2147483648 to +2147483647	Typical choice
SMALLINT	2 bytes	-32768 to +32767	Small integers
BIGINT	8 bytes	-9223372036854775808 to +9223372036854775807	Large integers
DECIMAL	variable	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point	user-defined precision
SERIAL	variable	1 to 2147483647	autoincrementing integer

Data Types

Numeric

Type	Storage size	Range	Notes
INT	4 bytes	-2147483648 to +2147483647	Typical choice
SMALLINT	2 bytes	-32768 to +32767	Small integers
BIGINT	8 b		
NUMERIC	var		

```
numeric(precision, scale)
```

Precision: total count of digits

Scale: count of decimal places

24.99 — 2 decimal places

```
numeric(4,2)
```

Data Types

Strings

Type	Storage size	Example	Notes
character varying(n), varchar(n)	variable-length with limit	Any text, "Hello"	Less flexible to change!
character(n), char(n)	fixed-length, blank padded	"M" or"F"	Not better in performance!
text	variable unlimited length	Any text, "Hello"	Winner!

Which one to choose?

Data Types

Strings

How about ZIP codes or phones numbers?

ZIP code: 0142

phone: 0049-234422

They don't have a numerical meaning!

Rather stored as string!

Data Types

Date/time

Type	Description	Example
`date`	Just date without time	`'2022-11-28'`
`time (with/without time zone)`	Just time without date	`'01:02:03.678'`
`timestamp (with/without time zone)`	Date and time	`'2022-11-28 01:02:03.678+02'`
`intervals`	Time interval	`'3 days 01:02:03.678'`

Data Types

Others

Type	Description	Example	Range
boolean	state of true or false	is_in_stock	TRUE, FALSE, NULL

Allowed input:

true	false
yes	no
1	0
on	off

Data Types

Others

Type	Description	Example	Range
boolean	state of true or false	is_in_stock	TRUE, FALSE, NULL
enum	A value of a list of ordered values	movie_rating	User-defined

```
CREATE TYPE mppa_rating AS ENUM ('G','PG', […])
```

Data Types

Others

Type	Description	Example	Range
boo			NULL
enu			
arr			type

name	phone
Peter	{'+48-4893245123', '+46-323245143'}
Frank	{'+41-39190643'}
Maya	{'+42-66764453', '+434567651234', '+43123676514'}

```
SELECT name, phone FROM customers
```

Data Types

Others

Type	Description	Example	Range
boo			NULL
enu			
arr			type

name	phone
Peter	+48-4893245123
Frank	+41-39190643
Maya	+42-66764453

```
SELECT name, phone[1] FROM customers
```

Data Types

Others

Type	Description	Example	Range
boo			NULL
enu			
array	Stores a list of values	text[] or int[]	Depending on type

name	phone
Maya	{'+42-66764453', '+434567651234', '+43123676514'}

```
SELECT name, phone[1] FROM customers
    WHERE '+42-66764453' = ANY (phones)
```

Constraints

- Column name
- Data type
- Constraints

Constraints

What is a constraint?

Defined when table is created

Used to define rules for the data in a table

Prevent insert of invalid data

Can be on column or table level

Constraints

COLUMN CONSTRAINTS

What constraints do we have?

NOT NULL	Ensures that a column cannot have a NULL value
UNIQUE	Ensures that all values in a column are different
DEFAULT	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	Ensures referential integrity (only values of another column can be used)
CHECK	Ensures that the values in a column satisfies a specific condition

Constraints

TABLE CONSTRAINTS

What constraints do we have?

```
PRIMARY KEY ( column [, ... ] )
UNIQUE ( column [, ... ] )
CHECK ( search_condition )
```

Primary & Foreign Key

PRIMARY KEY

One or multiple columns that uniquely identify each row in a table

UNIQUE

NOT NULL

	film_id [PK] integer	title text	description text
1	1	ACADEMY DINOSAUR	A Epic Drama of a
2	2	ACE GOLDFINGER	A Astounding Epis
3	3	ADAPTATION HOLES	A Astounding Refl

Primary & Foreign Key

FOREIGN KEY

A Column (or multiple) that refers to the primary in another table

REFERENCING table
CHILD table

plain Messages Notif

	payment_id integer	customer_id smallint	staff_id smallint			
1	16050	269	2			
2	16051	269	1	98	0.99	2020-01-25 16:
3	16052	269	2	678	6.99	2020-01-28 22:44:14.996577+01

Data Output Explain Messages Notifications

	customer_id [PK] integer	store_id smallint	first_name text	last_name text
1	1	1	MARY	SMITH
2	2	1	PATRICIA	JOHNSON
3	3	1	LINDA	

REFERENCED table
PARENT table

Primary & Foreign Key

Notes

1. Foreign key does not need to be unique
2. Primary key and foreign keys are usually the columns to join tables
3. Can be created also in table creation process

Data Output | Explain | Messages | Notifications

	payment_id integer	customer_id smallint	staff_id smallint	rental_id integer	amount numeric (5,2)	payment_date timestamp with time zone
1	16050	269	2	7	1.99	2020-01-24 22:40:19.996577+01
2	16051	269	1	98	0.99	2020-01-25 16:16:50.996577+01
3	16052	269	2	678	6.99	2020-01-28 22:44:14.996577+01

CREATE TABLE

```
CREATE TABLE <table_name>
```

CREATE TABLE

```
CREATE TABLE <table_name> (
column_name1 TYPE
)
```

CREATE TABLE

```
CREATE TABLE <table_name> (
column_name1 TYPE,
column_name2 TYPE
)
```

CREATE TABLE

```
CREATE TABLE staff(
column_name1 TYPE,
column_name2 TYPE
)
```

CREATE TABLE

```
CREATE TABLE staff(
staff_id INT,
column_name2 TYPE
)
```

CREATE TABLE

```
CREATE TABLE staff(
staff_id INT,
name VARCHAR(50)
)
```

CREATE TABLE

```
CREATE TABLE staff(
staff_id INT,
name VARCHAR(50)
)
```

CREATE TABLE

```
CREATE TABLE <table_name>(
 column_name1 TYPE [CONSTRAINT],
 column_name2 TYPE [CONSTRAINT],
 […])
```

CREATE TABLE

```
CREATE TABLE staff(
staff_id INT PRIMARY KEY,
name VARCHAR(50)
)
```

CREATE TABLE

```
CREATE TABLE staff(
staff_id SERIAL PRIMARY KEY,
name VARCHAR(50)
)
```

SERIAL: Creates an auto-increment sequence
(typically used with primary key)

CREATE TABLE

```
CREATE TABLE staff(
staff_id SERIAL PRIMARY KEY,
name VARCHAR(50) NOT NULL
)
```

CREATE TABLE

```
CREATE TABLE staff(
staff_id SERIAL PRIMARY KEY,
name VARCHAR(50) UNIQUE NOT NULL
)
```

CREATE TABLE

```
CREATE TABLE staff(
staff_id SERIAL PRIMARY KEY,
name VARCHAR(50) NOT NULL
UNIQUE(name,staff_id)
)
```

CREATE TABLE

```
DROP TABLE <table_name>
```

CREATE TABLE

```
DROP TABLE IF EXISTS <table_name>
```

CREATE TABLE

```
DROP TABLE IF EXISTS directors
```

Data Output | Explain | Messages | Notifications

```
NOTICE:  table "directors" does not exist, skipping
DROP TABLE
```

INSERT

```
INSERT INTO <table>
```

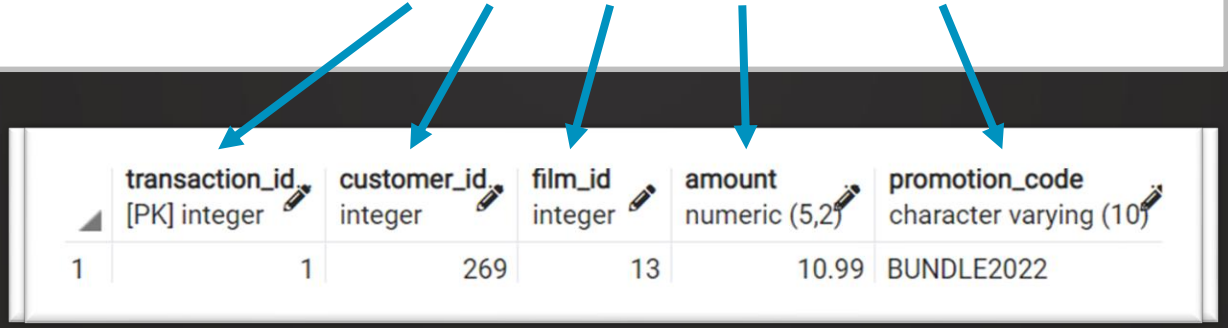
INSERT

```
INSERT INTO <table>
VALUES (value1,value2[,…])
```

transaction_id [PK] integer	customer_id integer	film_id integer	amount numeric (5,2)	promotion character varying (10)

INSERT

```
INSERT INTO online_sales
VALUES (1,269,13,10.99,'BUNDLE2022')
```



	transaction_id [PK] integer	customer_id integer	film_id integer	amount numeric (5,2)	promotion_code character varying (10)
1	1	269	13	10.99	BUNDLE2022

INSERT

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

	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

INSERT

```
INSERT INTO online_sales
(customer_id,amount)
VALUES (269,10.99)
```

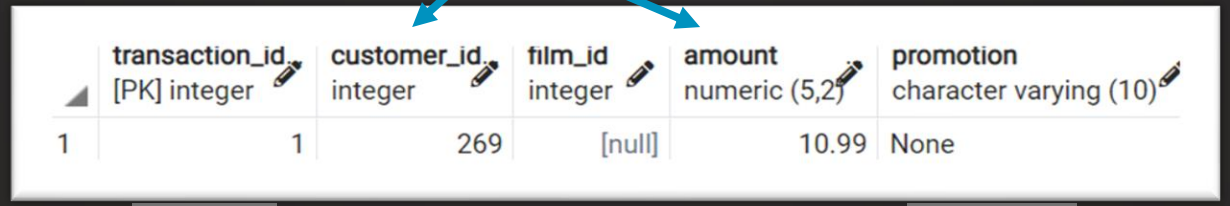
	transaction_id [PK] integer	customer_id integer	film_id integer	amount numeric (5,2)	promotion character varying (10)
1	1	269	[null]	10.99	None

SERIAL

DEFAULT

INSERT

```
INSERT INTO online_sales
(amount, customer_id)
VALUES (10.99,269)
```



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

SERIAL

DEFAULT

INSERT

```
INSERT INTO online_sales
(customer_id)
VALUES (269)
```

Data Output | Explain | Messages | Notifications

```
ERROR:  null value in column "amount" of relation "online_sales" violates not-null constraint
DETAIL:  Failing row contains (2, 269, null, null, None).
SQL state: 23502
```

INSERT

```
INSERT INTO online_sales
(customer_id, film_id,amount)
VALUES (269,13,10.99),(270,12,22.99)
```

	transaction_id [PK] integer	customer_id integer	film_id integer	amount numeric (5,2)	promotion character varying (10)
1	8	269	13	10.99	None
2	9	270	12	22.99	None

INSERT

```
INSERT INTO online_sales
(customer_id, film_id,amount)
VALUES
(269,13,10.99),
(270,12,22.99)
```

	transaction_id [PK] integer	customer_id integer	film_id integer	amount numeric (5,2)	promotion character varying (10)
1	8	269	13	10.99	None
2	9	270	12	22.99	None

ALTER TABLE

ADD, DELETE columns

ADD, DROP constraints

RENAME columns

ALTER data types

ALTER TABLE

```
ALTER TABLE <table_name>
ALTER_ACTION
```

ALTER TABLE

```
ALTER TABLE <table_name>
DROP COLUMN <column_name>
```

DROP

ALTER TABLE

```
ALTER TABLE staff
DROP COLUMN first_name
```

DROP

ALTER TABLE

```
ALTER TABLE staff
DROP COLUMN IF EXISTS first_name
```

DROP

ALTER TABLE

```
ALTER TABLE <table_name>
ADD COLUMN <column_name>
```

DROP

ADD

ALTER TABLE

```
ALTER TABLE staff
ADD COLUMN date_of_birth DATE
```

DROP

ADD

ALTER TABLE

```
ALTER TABLE staff
ADD COLUMN IF NOT EXISTS date_of_birth DATE
```

DROP

ADD

ALTER TABLE

```
ALTER TABLE <table_name>
ALTER COLUMN <column_name> TYPE NEW_TYPE
```

DROP

ADD

TYPE

ALTER TABLE

```
ALTER TABLE staff
ALTER COLUMN address_id TYPE SMALLINT
```

DROP

ADD

TYPE

ALTER TABLE

```
ALTER TABLE <table_name>
RENAME COLUMN <old_column_name> TO <new_column_name>
```

DROP

ADD

TYPE

RENAME

ALTER TABLE

```
ALTER TABLE staff
RENAME COLUMN first_name TO name
```

DROP

ADD

TYPE

RENAME

ALTER TABLE

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

DROP

ADD

TYPE

RENAME

DEFAULT

ALTER TABLE

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

DROP

ADD

TYPE

RENAME

DEFAULT

ALTER TABLE

```
ALTER TABLE staff
ALTER COLUMN store_id SET DEFAULT 1
```

DROP

ADD

TYPE

RENAME

DEFAULT

ALTER TABLE

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

DROP

ADD

TYPE

RENAME

DEFAULT

ALTER TABLE

```
ALTER TABLE <table_name>
ALTER COLUMN <column_name> SET NOT NULL
```

DROP

ADD

TYPE

RENAME

DEFAULT

ALTER TABLE

```
ALTER TABLE <table_name>
ALTER COLUMN <column_name> DROP NOT NULL
```

DROP

ADD

TYPE

RENAME

DEFAULT

NOT NULL

ALTER TABLE

```
ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name> UNIQUE(column1)
```

DROP

ADD

TYPE

RENAME

DEFAULT

NOT NULL

TABLE CONSTRAINT

ALTER TABLE

```
ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name>
UNIQUE(column1,column2[,…])
```

DROP

ADD

TYPE

RENAME

DEFAULT

NOT NULL

TABLE CONSTRAINT

ALTER TABLE

```
ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name>,
ADD PRIMARY KEY(column1,column2[,…])
```

DROP

ADD

TYPE

RENAME

DEFAULT

NOT NULL

TABLE CONSTRAINT

PRIMARY KEY

ALTER TABLE

```
ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name>,
ADD PRIMARY KEY(column1,column2[,…])
```

DROP

ADD

TYPE

RENAME

DEFAULT

NOT NULL

TABLE CONSTRAINT

PRIMARY KEY

ALTER TABLE

```
ALTER TABLE director
ALTER COLUMN director_account_name SET DEFAULT 3,
ALTER COLUMN first_name TYPE TEXT,
ALTER COLUMN last_name TYPE TEXT,
ADD COLUMN middle_name TEXT,
ADD CONSTRAINT constraint_1 UNIQUE(account_name)
```

DROP

ADD

TYPE

RENAME

DEFAULT

NOT NULL

TABLE CONSTRAINT

PRIMARY KEY

ALTER TABLE

```
ALTER TABLE director
ALTER COLUMN director_account_name SET DEFAULT 3,
ALTER COLUMN first_name TYPE TEXT,
ALTER COLUMN last_name TYPE TEXT,
ADD COLUMN middle_name TEXT,
ADD CONSTRAINT constraint_1 UNIQUE(account_name)
```

DROP

ADD

TYPE

RENAME

DEFAULT

NOT NULL

TABLE CONSTRAINT

PRIMARY KEY

ALTER TABLE

```
ALTER TABLE old_table_name
RENAME new_table_name
```

DROP

ADD

TYPE

RENAME

DEFAULT

NOT NULL

TABLE CONSTRAINT

PRIMARY KEY

ALTER TABLE

```
ALTER TABLE director
RENAME director_table
```

DROP

ADD

TYPE

RENAME

DEFAULT

NOT NULL

TABLE CONSTRAINT

PRIMARY KEY

CHECK

Limit the value range that can be placed in a column

CHECK

```
CREATE TABLE <table_name> (
<column_name> TYPE CHECK(condition))
```

CHECK

```
CREATE TABLE director (
name TEXT CHECK (length(name)>1))
```

```
ERROR:  new row for relation "director" violates check constraint "director_first_name_check"
DETAIL:  Failing row contains (1, null, M, null, null, null).
SQL state: 23514
```

CHECK

```
CREATE TABLE director(
name TEXT CONSTRAINT name_length CHECK (length(name)>1))
```

CHECK

```
CREATE TABLE director (
name TEXT CHECK (length(name)>1))
```

Default name: <table>_<column>_check

CHECK

```
CREATE TABLE director (
name TEXT CHECK (length(name)>1))
```

Default name: `<table>_<column>_check`

Default name: `director_name_check`

CHECK

```
CREATE TABLE director (
name TEXT,
date_of_birth DATE,
start_date DATE,
end_date DATE CHECK(start_date > '01-01-2000'))
```

Default name: `director_start_date_check`

CHECK

```
CREATE TABLE director (
name TEXT,
date_of_birth DATE,
start_date DATE,
end_date DATE CHECK(start_date > date_of_birth))
```

Default name: `director_check`

CHECK

```
CREATE TABLE director (
name TEXT,
date_of_birth DATE,
start_date DATE,
end_date DATE CHECK(start_date > date_of_birth))
```

```
INSERT INTO director
(date_of_birth,start_date)
VALUES ('01-01-1902','01-01-1900')
```

```
new row for relation "director" violates
check constraint "director_check"
```

CHECK

```
ALTER TABLE director
ADD CONSTRAINT date_check CHECK(start_date < end_date )
```

CHECK

```
ALTER TABLE director
DROP CONSTRAINT date_check
```

CHECK

```
ALTER TABLE director
RENAME CONSTRAINT date_check TO data_constraint
```

CHECK

```
CREATE TABLE director (
name TEXT,
date_of_birth DATE CHECK(data_of_birth >'01-01-1900'))
```

Challenge

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

song_id	song_name	genre	price	release_date
[PK] integer	character varying (30)	character varying (30)	numeric (4,2)	date

1. 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 insert a row like this:

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

6. Modify the constraint to be able to have 0.99 allowed as the lowest possible price.
7. Try again to insert the row.