Mathematical functions and operators

Operator	Description	Example	Result
+	addition	4 + 3	7
-	subtraction	5 - 3	2
*****	multiplication	4 * 2	8
/	division (integer division truncates the result)	8 / 4	2
%	modulo (remainder)	10 % 4	2
^	exponentiation	2 ^ 3	8

SYNTAX

```
SUM (replacement_cost) * 2
```

```
SUM (replacement_cost) + 1
```

```
SUM (replacement_cost) / SUM (rental_rate)*100
```

Mathematical functions and operators

Function	Description	Example	Result
abs(x)	absolute value	abs(-7)	7
round(x,d)	round x to d decimal places	round(4.3543)	4.35
ceiling(x)	round up to integer	ceiling(4.3543)	5
floor(x)	round down to integer	floor(4.3543)	4

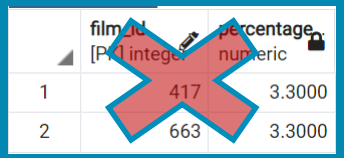
Challenge

Your manager is thinking about increasing the prices for films that are more expensive to replace.
For that reason, you should create a list of the films including the relation of rental rate / replacement cost where the rental rate is less than 4% of the replacement cost.

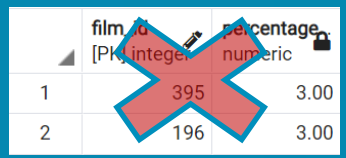
Create a list of that film_ids together with the percentage rounded to 2 decimal places. For example 3.54 *(=3.54%)*.

Result

	film_id [PK] integer	percentage numeric
1	417	3.30
2	663	3.30



	film_id [PK] integer	percentage numeric
1	417	3.3000
2	663	3.3000



	film_id [PK] integer	percentage numeric
1	395	3.00
2	196	3.00

CASE

Like IF/THEN statement:

Goes through a set of conditions returns a value if a condition is met

SYNTAX

Start of CASE statement

```
CASE
WHEN condition1 THEN result1
WHEN condition2 THEN result2
WHEN conditionN THEN resultN
ELSE result
END
```

Conditions & results

End of CASE statement

SYNTAX

```
SELECT
amount,
CASE
WHEN amount < 2 THEN 'low amount'
WHEN amount < 5 THEN 'medium amount'
ELSE 'high amount'
END
FROM payment
```

amount numeric (5,2)	case text
1.99	low amout
0.99	low amout
6.99	high amount
0.99	low amout
4.99	medium amount

SYNTAX

```
SELECT
TO_CHAR(book_date,'Dy'),
TO_CHAR(book_date,'Mon'),
CASE
WHEN TO_CHAR(book_date,'Dy')='Mon'THEN 'Monday special'
WHEN TO_CHAR(book_date,'Mon')='Jul' THEN 'July special'
END
FROM bookings
```

Result of first true condition!

Mon	Aug	Monday special
Sat	Jul	July special
Tue	Jul	July special
Mon	Jul	Monday special

SYNTAX

```
SELECT
TO_CHAR(book_date,'Dy'),
TO_CHAR(book_date,'Mon'),
CASE
WHEN TO_CHAR(book_date,'Dy')='Mon'THEN 'Monday special'
WHEN TO_CHAR(book_date,'Mon')='Jul' THEN 'July special'
END
FROM bookings
```

No condition met => [null]

Wed	Jul	July special
Fri	Jul	July special
Tue	Aug	[null]
Thu	Aug	[null]
Mon	Aug	Monday special

SYNTAX

```
SELECT
TO_CHAR(book_date,'Dy'),
TO_CHAR(book_date,'Mon'),
CASE
WHEN TO_CHAR(book_date,'Dy')='Mon'THEN 'Monday special'
WHEN TO_CHAR(book_date,'Mon')='Jul' THEN 'July special'
ELSE 'no special'
END
FROM bookings
```

ELSE => result if no condition is met

Wed	Jul	July special
Fri	Jul	July special
Tue	Aug	no special
Thu	Aug	no special
Mon	Aug	Monday special

SYNTAX

```
SELECT
total_amount,
TO_CHAR(book_date,'Dy'),
CASE
WHEN TO_CHAR(book_date,'Dy')='Mon'THEN 'Monday special'
WHEN total_amount < 30000 THEN 'Special deal'
ELSE 'no special at all'
END
FROM bookings
```

SYNTAX

```
SELECT
total_amount,
TO_CHAR(book_date,'Dy'),
CASE
WHEN TO_CHAR(book_date,'Dy')='Mon'THEN 'Monday special'
WHEN total_amount*1.4 < 30000 THEN 'Special deal'
ELSE 'no special at all'
END
FROM bookings
```

total_amount numeric (10,2)	to_char text	case text
265700.00	Wed	no special at all
37900.00	Fri	no special at all
18100.00	Tue	Special deal
131800.00	Thu	no special at all
23600.00	Mon	Monday special

SYNTAX

```
SELECT
total_amount,
TO_CHAR(book_date,'Dy'),
CASE
     WHEN TO_CHAR(book_date,'Dy')='Mon'THEN 'Monday special'
     WHEN total_amount < 30000 THEN 'Special deal'
     ELSE 'no special at all'
END
FROM bookings
```

Challenge

You need to find out how many tickets you have sold in the following categories:

- Low price ticket: total_amount < 20,000
- Mid price ticket: total_amount between 20,000 and 150,000
- High price ticket: total_amount >= 150,000

How many high price tickets has the company sold?

Result

Data Output | Explain | Mess

	count bigint	ticket_price text
1	205036	mid price ticket
2	30012	high price ticket
3	27740	low price ticket

Challenge

You need to find out how many flights have departed in the following seasons:

- Winter: December, January, Februar
- Spring: March, April, May
- Summer: June, July, August
- Fall: September, October, November

Result

Data Output | Explain

	flights bigint	season text
1	7596	Fall
2	25525	Summer

Challenge

You want to create a tier list in the following way:

1. Rating is 'PG' or 'PG-13' or length is more then 210 min:
   'Great rating or long (tier 1)
2. Description contains 'Drama' and length is more than 90min:
   'Long drama (tier 2)'
3. Description contains 'Drama' and length is not more than 90min:
   'Shcity drama (tier 3)'
4. Rental_rate less than $1:
   'Very cheap (tier 4)'

If one movie can be in multiple categories it gets the higher tier assigned.
How can you filter to only those movies that appear in one of these 4 tiers?

Result

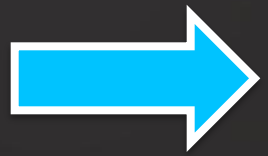
	title text	case text
1	ACADEMY DINOSAUR	Great rating or very long (tier 1)
2	AGENT TRUMAN	Great rating or very long (tier 1)
3	AIRPLANE SIERRA	Great rating or very long (tier 1)
4	ALABAMA DEVIL	Great rating or very long (tier 1)
5	ALAMO VIDEOTAPE	Very cheap (tier 4)

COALESCE

✓ **Returns first value of a list of values which is not null**

COALESCE(actual_arrival, schedule_arrival)

actual_arrival timestamp with time zone	scheduled_arrival timestamp with time zone
2017-08-05 19:01:00+02	2017-08-05 19:00:00+02
2017-08-05 09:34:00+02	2017-08-05 09:30:00+02
[null]	2017-09-09 12:20:00+02



coalesce timestamp with time zone
2017-08-05 19:01:00+02
2017-08-05 09:34:00+02
2017-09-09 12:20:00+02

SYNTAX

```
COALESCE (actual_arrival, scheduled_arrival)
```

SYNTAX

```
COALESCE (actual_arrival, '1970-01-01 0:00')
```

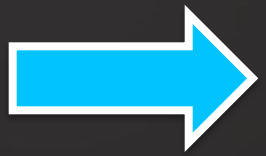
actual_arrival timestamp with time zone	scheduled_arrival timestamp with time zone	coalesce timestamp with time zone
[null]	2017-09-10 13:55:00+02	1970-01-01 00:00:00+01

CAST

✓ **Changes the data type of a value**

scheduled_arrival timestamp with time zone
2017-09-10 13:55:00+02
2017-08-25 16:35:00+02

CAST TO TEXT



scheduled_arrival character varying
2017-09-10 13:55:00+02
2017-08-25 16:35:00+02

SYNTAX

```
CAST (value/column AS data type)
```

SYNTAX

```
CAST (scheduled_arrival AS data type)
```

SYNTAX

```
CAST (scheduled_arrival AS VARCHAR)
```

scheduled_arrival character varying
2017-09-10 13:55:00+02
2017-08-25 16:35:00+02

SYNTAX

```
CAST (scheduled_arrival AS DATE)
```

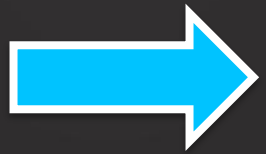
scheduled_arrival timestamp with time zone	scheduled_arrival date
2017-09-10 13:55:00+02	2017-09-10
2017-08-25 16:35:00+02	2017-08-25
2017-09-05 13:15:00+02	2017-09-05

REPLACE

✓ **Replaces text from a string in a column with another text**

flight_no character (6)
PG0134
PG0052
PG0561

REPLACE 'PG' with 'FL'



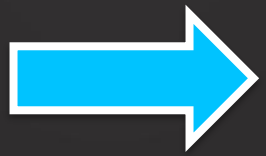
replace text
FL0134
FL0052
FL0561

REPLACE

✓ **Replaces text from a string in a column with another text**

flight_no character (6)
PG0134
PG0052
PG0561

REPLACE 'PG' with ''



replace text
0134
0052
0561

SYNTAX

```
REPLACE (column, old_text, new_text)
```

SYNTAX

```
REPLACE (flight_no, 'PG', 'FL')
```

replace text
FL0134
FL0052
FL0561

SYNTAX

```
REPLACE (flight_no, 'PG', '')
```

replace text
0134
0052
0561