Window Functions

emp_no	department	salary
8	sales	59000
12	sales	60000
20	customer service	56000
21	customer service	55000

```
SELECT department, AVG(salary) FROM emps GROUP BY department;
```

```
+--------+-----------------+--------+
| emp_no | department      | salary |
+--------+-----------------+--------+
|      8 | sales           |  59000 |
|     12 | sales           |  60000 |
|     20 | customer service |  56000 |
|     21 | customer service |  55000 |
+--------+-----------------+--------+
```

'sales' group

```
|      8 | sales           |  59000 |
|     12 | sales           |  60000 |
```

'customer service' group

```
|     20 | customer service |  56000 |
|     21 | customer service |  55000 |
```

```
+-----------------+-------------+
| department      | AVG(salary) |
+-----------------+-------------+
| sales           |  59500.0000 |
| customer service |  55500.0000 |
+-----------------+-------------+
```

GROUP BY

```
SELECT department, AVG(salary) FROM emps GROUP BY department;
```

```
+------------------+-------------+
| department       | AVG(salary) |
+------------------+-------------+
| sales            |  59500.0000 |
| customer service |  55500.0000 |
+------------------+-------------+
```

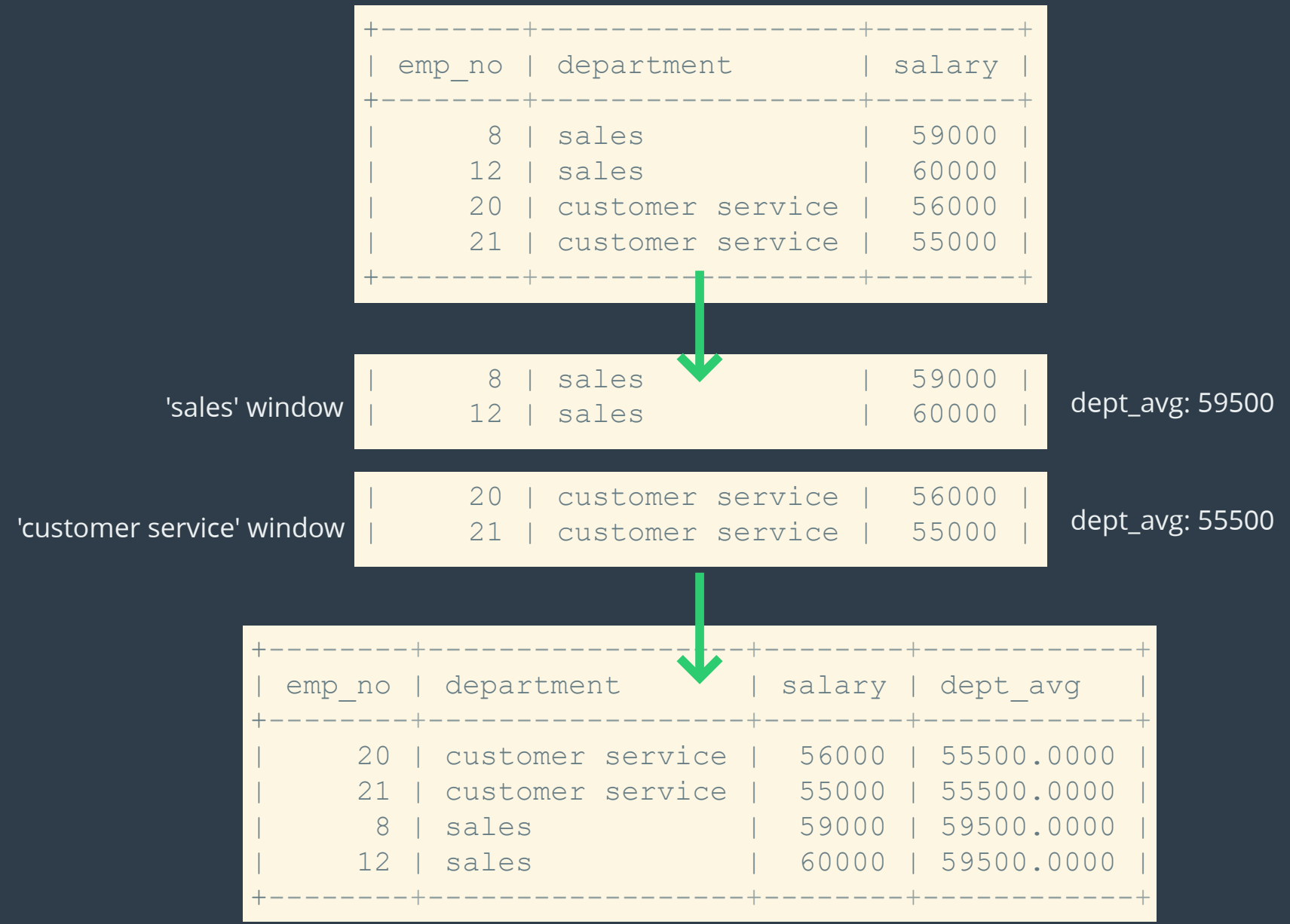
GROUP BY collapses each group of rows

INTO a single result row

WINDOW FUNCTIONS

Window functions perform aggregate operations on groups of rows, but they produce a result FOR EACH ROW.

```
SELECT emp_no, department, salary,
       AVG(salary) OVER(PARTITION BY department) AS dept_avg FROM emps;
```



OVER

```
AVG(salary) OVER()
```

The OVER() clause constructs a window. When it's empty, the window will include all records

PARTITION BY

```
AVG(salary) OVER(PARTITION BY department)
```

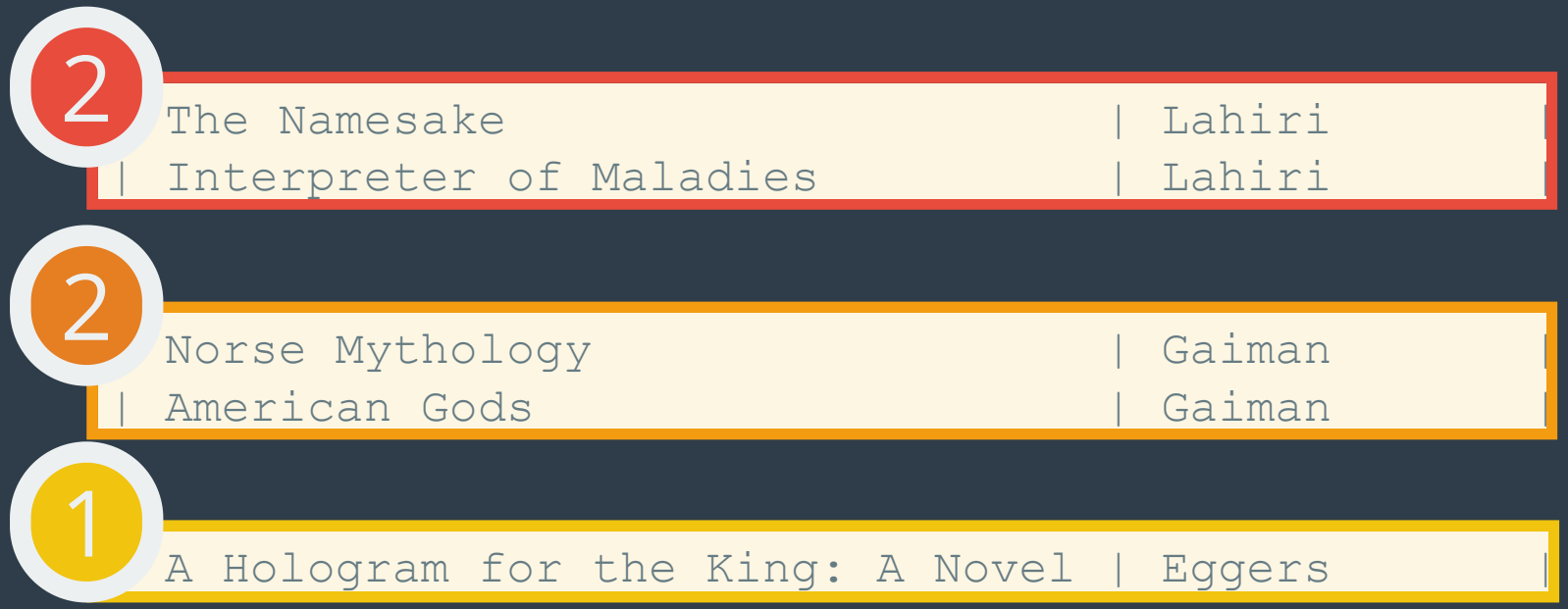
Inside of the the OVER(), use PARTITION BY to form rows into groups of row

ORDER BY

```
OVER(ORDER BY salary DESC)
```

Use ORDER BY inside of the OVER() clause to re-order rows within each window.

```sql
SELECT author_lname, COUNT(*)
FROM books GROUP BY author_lname;
```



UNIQUE constraint

```
CREATE TABLE companies (
    supplier_id INT AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    phone VARCHAR(15) NOT NULL UNIQUE,
    address VARCHAR(255) NOT NULL,
    PRIMARY KEY (supplier_id)
);
```

phone must be unique

CHECK constraints

```
CREATE TABLE partiers (
  name VARCHAR(50),
  age INT CHECK (age > 18)
);
```

age must be greater than 18

Named Constraints

```sql
CREATE TABLE partiers2 (
  name VARCHAR(50),
  age INT,
  CONSTRAINT age_over_18 CHECK (age > 18)
);
```

We can provide a name for the constraint

Multi-Column Checks

```
CREATE TABLE companies (
    supplier_id INT AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    phone VARCHAR(15) NOT NULL UNIQUE,
    address VARCHAR(255) NOT NULL,
    PRIMARY KEY (supplier_id),
    CONSTRAINT name_address UNIQUE (name , address)
);
```

The combination of name and address must be unique for each row

ALTER TABLE

```
ALTER TABLE companies
ADD COLUMN city VARCHAR(25);
```

Use ALTER TABLE to add a new column to an existing table

ALTER TABLE

```
ALTER TABLE suppliers
DROP COLUMN city;
```

Use ALTER TABLE to remove
columns from a table

Renaming Tables

```
RENAME TABLE companies TO suppliers;
```

Use RENAME to...rename a table

Renaming Columns

```
ALTER TABLE suppliers
RENAME COLUMN name TO biz_name;
```

Use RENAME COLUMN to rename a column

Changing Columns

```
ALTER TABLE suppliers
CHANGE business biz_name VARCHAR(50);
```

Use CHANGE to rename a column AND change its data type.

This example renames the 'business' column to 'biz_name'
AND make its VARCHAR(50)

Modify

```
ALTER TABLE suppliers
MODIFY biz_name VARCHAR(100);
```

Use MODIFY to change an existing column's type

This example modify's the 'biz_name' column to be a VARCHAR(100)

NOT LIKE

```
SELECT title FROM books
WHERE title NOT LIKE 'W%';
```

Select books with titles that don't start with 'W'

Greater Than

```sql
SELECT * FROM books
WHERE released_year > 2000;
```

Select books released after the year 2000

A Small Side Note

```
SELECT 99 > 1;
```

What Do You Expect?

Less Than

```sql
SELECT * FROM books
WHERE released_year < 2000;
```

Select books released **before** the year 2000

Less Than Or Equal To

LOGICAL AND

SELECT books written
by Dave Eggers,
published after the
year 2010

SELECT books written
by Dave Eggers,
published after the
year 2010

SELECT books written by Dave Eggers,

```
SELECT * FROM books
WHERE author_lname='Eggers';
```

```
SELECT * FROM books
WHERE released_year > 2010;
```

published after the year 2010

```sql
SELECT * FROM books
WHERE author_lname='Eggers';
```

AND

```sql
SELECT * FROM books
WHERE released_year > 2010;
```

```sql
SELECT * FROM books
WHERE author_lname='Eggers' AND
released_year > 2010;
```

```
SELECT * FROM books
WHERE author_lname='Eggers' AND
released_year > 2010 AND
title LIKE '%novel%';
```

OR

LOGICAL OR

Let's Try Something!

```
SELECT * FROM books
WHERE author_lname='Eggers' AND
released_year > 2010;
```

```
SELECT * FROM books
WHERE author_lname='Eggers' OR
released_year > 2010;
```

Condition 1 AND Condition 2

BOTH SIDES MUST BE TRUE

Condition 1 OR Condition 2

ONLY ONE SIDE MUST BE TRUE

BETWEEN

Before we see BETWEEN...

How can we accomplish the same thing using what we already know?

Select all books published between 2004 and 2015

USE LOGICAL AND!

```
SELECT title, released_year FROM
books WHERE released_year >= 2004 AND
released_year <= 2015;
```

Or...Use BETWEEN

BETWEEN x AND y

```
SELECT title, released_year FROM books
WHERE released_year BETWEEN 2004 AND 2015;
```

NOT BETWEEN

Is Also A Thing

NOT BETWEEN x AND y

```
SELECT title, released_year FROM books
WHERE released_year NOT BETWEEN 2004 AND 2015;
```

A Note About Comparing Dates

IN

Select all books written by...

Carver

OR...

Lahiri

OR...

Smith

We Can Already Do This

```
SELECT title, author_lname FROM books
WHERE author_lname='Carver' OR
      author_lname='Lahiri' OR
      author_lname='Smith';
```

IN makes it much easier

SO MUCH SHORTER!

```
SELECT title, author_lname FROM books
WHERE author_lname IN ('Carver', 'Lahiri', 'Smith');
```

NOT IN

I'm really superstitious and have a problem with even numbers...

(I'm not actually, I promise)

Select all books not published in

- 2000,
- 2002,
- 2004,
- 2006,
- 2008,
- 2010,
- 2012,
- 2014,
- 2016

Well...that was fun to type

```
SELECT title, released_year FROM books
WHERE released_year != 2000 AND
      released_year != 2002 AND
      released_year != 2004 AND
      released_year != 2006 AND
      released_year != 2008 AND
      released_year != 2010 AND
      released_year != 2012 AND
      released_year != 2014 AND
      released_year != 2016;
```

There's a better way!

```sql
SELECT title, released_year FROM books
WHERE released_year NOT IN
(2000,2002,2004,2006,2008,2010,2012,2014,2016);
```

Taking It To The Next Level!

I only want books released after 2000

```sql
SELECT title, released_year FROM books
WHERE released_year >= 2000
AND released_year NOT IN
(2000,2002,2004,2006,2008,2010,2012,2014,2016);
```

Theres still a better way

It doesn't use IN or NOT IN

%

MODULO

```
SELECT title, released_year FROM books
WHERE released_year >= 2000 AND
released_year % 2 != 0;
```

CASE STATEMENTS

title	released_year	GENRE
The Namesake	2003	Modern Lit
Norse Mythology	2016	Modern Lit
American Gods	2001	Modern Lit
Interpreter of Maladies	1996	20th Century Lit
A Hologram for the King: A Novel	2012	Modern Lit
The Circle	2013	Modern Lit
The Amazing Adventures of Kavalier & Clay	2000	Modern Lit
Just Kids	2010	Modern Lit
A Heartbreaking Work of Staggering Genius	2001	Modern Lit
Coraline	2003	Modern Lit
What We Talk About When We Talk About Love: Stories	1981	20th Century Lit
Where I'm Calling From: Selected Stories	1989	20th Century Lit
White Noise	1985	20th Century Lit
Cannery Row	1945	20th Century Lit
Oblivion: Stories	2004	Modern Lit
Consider the Lobster	2005	Modern Lit
10% Happier	2014	Modern Lit
fake_book	2001	Modern Lit
Lincoln In The Bardo	2017	Modern Lit

```sql
SELECT title, released_year,
       CASE
         WHEN released_year >= 2000 THEN 'Modern Lit'
         ELSE '20th Century Lit'
       END AS GENRE
FROM books;
```

title	stock_quantity	STOCK
The Namesake	32	*
Norse Mythology	43	*
American Gods	12	*
Interpreter of Maladies	97	**
A Hologram for the King: A Novel	154	***
The Circle	26	*
The Amazing Adventures of Kavalier & Clay	68	**
Just Kids	55	**
A Heartbreaking Work of Staggering Genius	104	***
Coraline	100	**
What We Talk About When We Talk About Love: Stories	23	*
Where I'm Calling From: Selected Stories	12	*
White Noise	49	*
Cannery Row	95	**
Oblivion: Stories	172	***
Consider the Lobster	92	**
10% Happier	29	*
fake_book	287	***
Lincoln In The Bardo	1000	***

```sql
SELECT title, stock_quantity,
    CASE
        WHEN stock_quantity BETWEEN 0 AND 50 THEN '*'
        WHEN stock_quantity BETWEEN 51 AND 100 THEN '**'
        ELSE '***'
    END AS STOCK
FROM books;
```

A Little More Succinct

```
SELECT title, stock_quantity,
    CASE
        WHEN stock_quantity <= 50 THEN '*'
        WHEN stock_quantity <= 100 THEN '**'
        ELSE '***'
    END AS STOCK
FROM books;
```

YOUR TURN

This is the last time we'll work with the books table!!!

Evaluate the following...

```
SELECT 10 != 10;
```

```
SELECT 15 > 14 AND 99 - 5 <= 94;
```

```
SELECT 1 IN (5,3) OR 9 BETWEEN 8 AND 10;
```

Select All Books Written Before 1980 (non inclusive)

Select All Books Written By Eggers Or Chabon

Select All Books Written By Lahiri, Published after 2000

Select All books with page counts between 100 and 200

Select all books where author_lname starts with a 'C' or an 'S"

If title contains 'stories' -> Short Stories

Just Kids and A Heartbreaking Work -> Memoir

Everything Else -> Novel

title	author_lname	TYPE
The Namesake	Lahiri	Novel
Norse Mythology	Gaiman	Novel
American Gods	Gaiman	Novel
Interpreter of Maladies	Lahiri	Novel
A Hologram for the King: A Novel	Eggers	Novel
The Circle	Eggers	Novel
The Amazing Adventures of Kavalier & Clay	Chabon	Novel
Just Kids	Smith	Memoir
A Heartbreaking Work of Staggering Genius	Eggers	Memoir
Coraline	Gaiman	Novel
What We Talk About When We Talk About Love: Stories	Carver	Short Stories
Where I'm Calling From: Selected Stories	Carver	Short Stories
White Noise	DeLillo	Novel
Cannery Row	Steinbeck	Novel
Oblivion: Stories	Foster Wallace	Short Stories
Consider the Lobster	Foster Wallace	Novel
10% Happier	Harris	Novel
fake_book	Harris	Novel
Lincoln In The Bardo	Saunders	Novel

Bonus: Make This Happen

author_fname	author_lname	COUNT
Jhumpa	Lahiri	2 books
Neil	Gaiman	3 books
Dave	Eggers	3 books
Michael	Chabon	1 book
Patti	Smith	1 book
Raymond	Carver	2 books
Don	DeLillo	1 book
John	Steinbeck	1 book
David	Foster Wallace	2 books
Dan	Harris	1 book
Freida	Harris	1 book
George	Saunders	1 book

BURN ALL THE BOOKS

We're Done With Them!